

Robust Local Preserving and Global Aligning Network for Adversarial Domain Adaptation

Wenwen Qiang*, Jiangmeng Li*, Changwen Zheng, Bing Su, and Hui Xiong, *Fellow, IEEE*

Abstract—Unsupervised domain adaptation (UDA) requires source domain samples with clean ground truth labels during training. Accurately labeling a large number of source domain samples is time-consuming and laborious. An alternative is to utilize samples with noisy labels for training. However, training with noisy labels can greatly reduce the performance of UDA. In this paper, we address the problem that learning UDA models only with access to noisy labels and propose a novel method called robust local preserving and global aligning network (RLPGA). RLPGA improves the robustness of the label noise from two aspects. One is learning a classifier by a robust informative-theoretic-based loss function. The other is constructing two adjacency weight matrices and two negative weight matrices by the proposed local preserving module to preserve the local topology structures of input data. We conduct theoretical analysis on the robustness of the proposed RLPGA and prove that the robust informative-theoretic-based loss and the local preserving module are beneficial to reduce the empirical risk of the target domain. A series of empirical studies show the effectiveness of our proposed RLPGA.

Index Terms—Wasserstein distance, unsupervised domain adaptation, noisy label, representation learning, adversarial learning.

1 INTRODUCTION

UNSUPERVISED domain adaptation emphasizes [1], [2], [3] the problem of learning a classifier that can be transferred across two domains. In general, the samples in the source domain are labeled, while the samples in the target domain are unlabeled. The main challenge in this research area is to reduce the difference between the probability distributions of two domains [4], [5], [6], [7]. To this end, the strategy based on discrepancy minimization has attracted much attention. Among them, adversarial learning methods have achieved remarkable performance improvements [8].

The training set of unsupervised adversarial domain adaptation models consists of two parts including the labeled source domain samples and unlabeled target domain samples. However, it is usually very expensive and tedious to accurately label large source domain training samples. An alternative way is to collect labels of samples from some crowdsourcing platforms in which the cost is cheaper and

easier but the obtained labels are always contaminated by noise. As a result, the performance of adversarial domain adaptation models learning from noisy labels will be decreased. One reason is that adversarial domain adaptation models usually learn the classifier by minimizing the cross-entropy loss. The cross-entropy loss can be regarded as the distance between the outputs of the classifier and the labels, so it is sensitive to label noises. That is to say, when a careless annotator tends to label positive class to negative class, then the distance-based loss would force adversarial domain adaptation to learn a classifier who is more likely to output negative class than to output true label.

However, to make the domain adaptive models robust to label noises, it is not enough to only employ robust classification loss to train the models. The main reason is that robust loss can only reduce the impact of noisy labels, but can not completely eliminate it. Actually, for the source domain, learning with noisy labels can reduce the feature discriminability of samples in the latent space. This can lead to that a sample belonging to the one class is easy to be misclassified into another class. From a geometrically intuitive point of view, if a sample belonging to class i is incorrectly labeled as class j , then the gradient back-propagation operation will force the sample to go from a place surrounded by many samples of the same type to a place surrounded by many different class samples. Therefore, besides learning based on a robust classification loss, designing an unsupervised method to maintain the local structure of the data distribution is also very important.

To tackle these issues, we propose a novel method for adversarial domain adaptation named *Robust Local Preserving and Global Aligning Network* (RLPGA). RLPGA consists of three parts including a robust loss function for learning a classifier, a local preserving module, and a global aligning module. Firstly, RLPGA projects samples of both domains into a latent space. Then, a robust informative-theoretic-

*W. Qiang and J. Li are with the University of Chinese Academy of Sciences, Beijing, China. They are also with the Science & Technology on Integrated Information System Laboratory, Institute of Software Chinese Academy of Sciences, Beijing, China. E-mail: a01114115@163.com, Jiangmeng2019@iscas.ac.cn

C. Zheng is with the Science & Technology on Integrated Information System Laboratory, Institute of Software Chinese Academy of Sciences, Beijing, China. E-mail: changwen@iscas.ac.cn

B. Su is with the Beijing Key Laboratory of Big Data Management and Analysis Methods, Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, 100872, China. E-mail: subingats@gmail.com

H. Xiong is with the Hong Kong University of Science and Technology (Guangzhou). E-mail: xionghui@ust.hk.

*They have contributed equally to this work (Corresponding author: Bing Su).

©2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

based loss function is minimized to learn the classifier. The global aligning module minimizes the Wasserstein distance between source domain distribution and target domain distribution. The local preserving module constructs two adjacency weight matrices and two negative weight matrices to encode the local topological relationship among samples and propose an objective function based on the graphs for local preserving. The major contributions of this paper are three-fold:

- A robust informative-theoretic-based loss function is proposed to measure the performance of the classifier for adversarial domain adaptation.
- To reduce the effect of label noises from the perspective of the learned feature representation, we propose a new objective function for preserving local neighbor topology based on constructing two adjacency weight matrices and two negative weight matrices. We jointly minimize the Wasserstein distance between two domain distributions and the new objective. In this way, the margins between different classes are enlarged and hence the learned features are more discriminative and robust.
- We provide theoretical analysis on the robustness of RLPGA, and prove that the robust loss and the enhanced feature discriminability are beneficial to reduce the empirical risk in the target domain.

2 RELATED WORKS

Domain Adaptation Algorithm. Existing domain adaptation methods can be divided into three categories. The first is instance-based methods [9], [10], [11], [12], which enhance feature transferability by reweighting or subsampling the source domain samples. The second is parameter-based methods [13], [14], [15], [16], [17], which enhance the feature transferability by regularized terms or reweighting techniques. The last is representation learning based methods [18], [19], [20], [21], [22], which first learn a latent space, and then align feature distributions across domains based on the learned feature representation by two strategies.

For representation learning based methods, The first strategy to align feature distributions across domains is that moment matching based on statistical characteristics [23], [24]. *E.g.*, Maximum mean discrepancy (MMD) [25], [26] measures the divergence of two distributions in the reproducing kernel Hilbert space (RKHS) with the advantages that it can approximate any moment of the distribution by choosing a suitable kernel function. Deep correlation alignment (DCORAL) [27], [28] aligns two distributions by minimizing the difference in the second-order statistics of the two distributions. The other strategy is adversarial learning based on a zero-sum two-player game [29]. These adversarial methods have achieved remarkable performance improvements. The metric for adversarial learning based methods can be KL-divergence, H-divergence, and Wasserstein distance [1], [8], [18], [19], [30], [31], [32], [33]. Among them, Wasserstein distance takes advantage of gradient superiority. *E.g.*, Wasserstein Distance Guided Representation Learning (WDGRL) [8] aligns the two distributions by minimizing the Wasserstein distance, which takes the advantage of gradient superiority. Sliced Wasserstein Discrepancy

(SWD) [33] is a method that based on sliced Wasserstein distance. As for domain adaptation over noisy labels, there are few pioneering works [34], [35] to handle this problem, and most of them focus on a robust classifier loss. [36] proposes to tackle the label noise problem by clustering-based UDA methods for person re-ID. RLPGA handles robustness by considering both a robust classification loss and an enhanced feature discriminability.

Domain Adaptation Theory. There are rich advances in domain adaptation theory. A rigorous error bound for unsupervised domain adaptation is proposed by [37], [38]. Then, many extensions based on these bounds, from loss functions to Bayesian settings and regression problems, are put forward [13], [39], [40], [41], [42]. *E.g.*, Kuroki [43] proposes a discrepancy measure called S-disc, which can not only provide a tighter generalization error bound but also have a convergence guarantee. Germain [40] proposed a PAC-Bayesian theory based on the domain disagreement pseudometric. Another related work is about the Wasserstein distance based domain adaptation algorithm and proves that Wasserstein distance can guarantee generalization for domain adaptation [8]. As for our proposed method, the theoretical analysis for Wasserstein distance based domain adaptation can be directly extended to RLPGA. Also, RLPGA gives some theoretical analysis for robustness and advantage of enhancing the feature discriminability.

3 PRELIMINARIES

3.1 Problem definition

This paper considers the classification task of unsupervised domain adaptation (UDA). Let \mathcal{X} represent the input feature space and $\eta : \mathcal{X} \rightarrow \mathcal{Y}$ be the domain-invariant ground truth labeling function, where $X \in \mathcal{X}$, and Y is the label. Let P_s be the input distribution over X for the source domain and P_t be the input distribution over X for the target domain. Let \mathcal{Z} be a latent space and $F : \mathcal{X} \rightarrow \mathcal{Z}$ be a class of feature extractors, where $Z \in \mathcal{Z}$. For a domain $u \in \{s, t\}$, $P_u^{f(X)}(Z) = P_u(f^{-1}(Z))$ represents the induced probability distribution over \mathcal{Z} , where $f \in F$. For a given $Z \in \mathcal{Z}$. Denote $H : Z \rightarrow Y$ as a class of prediction functions. Then, the learned classifier can be represented as $h(f(X))$, where $h \in H$. The goal is to learn a classifier that can minimize the following expected target risk:

$$R_{P_t}(f, h) = \int P_t(X) |\eta(X) - h(f(X))| dX. \quad (1)$$

The difference between the supervised domain adaption (SDA) and UDA is that the label and the feature of the target domain dataset are all available during training phase for SDA, but for the UDA, we can only access to the feature of the target domain dataset during the training phase. For practical application, Dou [44] propose to utilize the adversarial learning to UDA from the source Medical Image Analysis (MIA) domain to the target Computed Tomography (CT) domain.

Generally, UDA models learn a classifier h using source domain samples with their ground truth labels. For real-world applications, it is costly to obtain the true data labels. Instead, we focus on using noisy labels to learn the classifier.

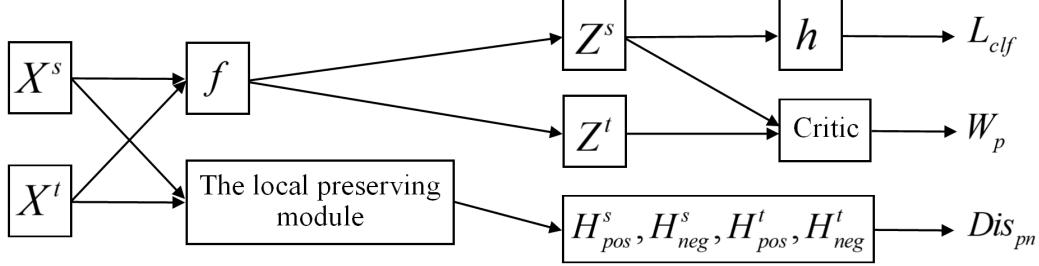


Fig. 1: The framework of RLPGA. X^s and X^t are the input datasets, f is the feature extractor, Z^s and Z^t are the feature representations, h is the classifier. First, RLPGA constructs four weight matrices by the local preserving module and maps the input samples to the latent space through the feature extractor. Then, RLPGA feeds Z^s into h to calculate the classification loss L_{clf} , feeds Z^s and Z^t into the Critic to calculate the Wasserstein distance between two domain distributions, and calculate the loss Dis_{pn} based on the four weight matrices. Then, RLPGA minimizes the equation 15 to update the parameters of f , h , and the Critic.

We denote the source domain data as X^s and its corresponding labels as Y^{s*} , where Y^{s*} is the noisy version of the clean ground truth labels Y^s . Therefore, our goal is to learn a robust classifier based on the source domain data with noisy labels, which can be transferred to the target domain.

Formally, we denote Y^* as the noisy version of Y and T_{YY^*} as the transition distribution from Y to Y^* . Let $Y, Y^* \in \{1, 2, \dots, C\}$, and C be the total number of the categories. Then we denote $\Pr(Y^* = j | Y = i)$ as the transition distribution to transfer the ground truth label i to the class j , where $i, j \in \{1, 2, \dots, C\}$. We use $T_{Y \rightarrow Y^*}$ to represent the $C \times C$ matrix, and $T_{Y \rightarrow Y^*}(i, j) = \Pr(Y^* = j | Y = i)$. Generally, the label noise can be defined into several kinds based on $T_{Y \rightarrow Y^*}$ [45], e.g., define the label noise as class independent (or uniform), then $T_{Y \rightarrow Y^*}$ can be written as $T_{Y \rightarrow Y^*}(i, j) = a$ if $i \neq j$ and $T_{Y \rightarrow Y^*}(i, j) = b$ if $i = j$, where $a, b > 0$ and $(C - 1)a + b = 1$. For real-life data, although the type of label noise is complicated, its $T_{Y \rightarrow Y^*}$ can be estimated through empirical distribution. Then, we have

Assumption 3.1. Assume the Markov chain: $X \rightarrow Y \rightarrow Y^*$ is hold. i.e., X is independent of Y^* conditioning on Y . Assume the transition distribution matrix $T_{Y \rightarrow Y^*}$ is invertible. i.e., $\det(T_{Y \rightarrow Y^*}) \neq 0$.

From the Assumption 3.1, we can know that $X \perp\!\!\!\perp Y^* | Y$. The invertible transition distribution matrix is to emphasize that the any class in $\{1, 2, \dots, C\}$ can be polluted by noise, and each class can be transferred to every C class with a certain possibility. This assumption is also to simulate the fact that every real label in the actual situation can be artificially incorrectly labeled as other classes.

3.2 Representation leaning-based domain adaptation

This paper focus on learning a domain-invariant representation for domain adaptation. The objective of representation learning-based domain adaptation methods is composed of three components [19], [21]. First, all training samples is mapped into the latent space \mathcal{Z} to obtain the feature representation by a projection function (or feature extractor) f . The first is to minimize a metric which measures the difference between two probability distributions to align the distributions of two domains. The second part is to minimizing the source domain classification risk. The third

is to minimize a regularization term. In short, the objective is formulated as

$$\min_{f, h} R_{P_s}(f, h) + \alpha D(P_s^{f(X)}, P_t^{f(X)}) + \Delta(f, h), \quad (2)$$

where $D(\cdot)$ is the metric that measures the difference between two domain distributions, Δ is a regularization term to punish the parameters of the feature extractor f and the classifier h , α is the corresponding hyperparameters, R_{P_s} is the risk of the classifier h over source domain samples based on the learned latent space Z :

$$R_{P_s}(f, h) = \int P_s(X) |\eta(X) - h(f(X))| dX. \quad (3)$$

In our setting, we only have access to the source domain samples with noisy labels. The aim is to learn an expected classifier h^* that is robust to label noises. Thus, in our setting, the equation 3 is written as

$$R_{P_s}(f, h) = \int P_s(X) |Y^* - h(f(X))| dX. \quad (4)$$

4 PROPOSED METHOD

In this section, we introduce the proposed robust local preserving and global aligning network (RLPGA). RLPGA consists of four parts. The first part is the feature extractor f that mapping the input dataset into the latent space to obtain the feature representation. The second part is the classifier to be learned based on the noisy labels of the source domain dataset. The third part is the local preserving module, which is to improve the robustness to the label noise based on preserving the local topological structure of the data in the input space. The fourth part is the global aligning module, which is to align the distributions of the source domain and the target domain. The overall framework of RLPGA is shown in Fig. 1.

4.1 The informative-theoretic-based loss for classifier

To learn a classifier h with the noisy label for UDA, we employ the determinant based mutual information (DMI) [46] to measure the performance of the classifier. We maximize the DMI between the the output of h and the noisy label Y^* . From the [46], [47], we can obtain the follows:

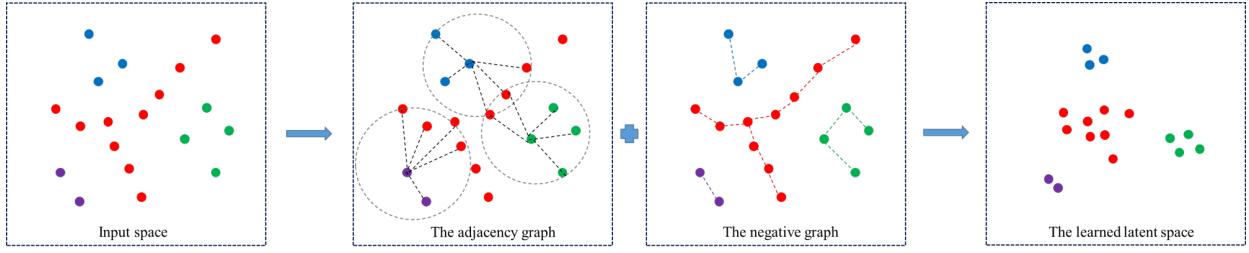


Fig. 2: A motivating example for the local preserving module. It contains an example for the adjacency graph and an example for the negative graph. We set $k = 3$ in construct the adjacency graph. The number of the finally obtained clusters in the negative graph is 4.

Definition 4.1. (Determinant based Mutual Information) For two discrete random variables X_1, X_2 , the determinant based mutual information between them is defined as

$$\text{DMI}(X_1, X_2) = |\det(\mathbf{T}_{X_1, X_2})|, \quad (5)$$

where \mathbf{T}_{X_1, X_2} is the matrix format of the joint distribution $\Pr(X_1, X_2)$ over X_1 and X_2 .

Therefore, let Y^* be the one-hot vector, the measurement for the performance of the classifier h is presented as $\text{DMI}(h(\cdot), Y^*)$. As we can see, calculating $\text{DMI}(h(\cdot), Y^*)$ need to obtain the joint distribution $\Pr(h(\cdot), Y^*)$. Because h is also a random variable, therefore, we can obtain:

$$\begin{aligned} \Pr(h(\cdot), Y^*) &= \int_X \Pr(h(\cdot), Y^*, X) d(X) \\ &= \int_X \Pr(h(\cdot), Y^* | X) P(X) d(X) \quad (6) \\ &= \int_X \Pr(h(X), Y^*(X)) P(X) d(X) \end{aligned}$$

Also, we can obtain the Markov chain: $h(X) \leftarrow X \rightarrow Y^*$. So, we can see: $h(X) \perp\!\!\!\perp Y^* | X$. Thus, during training process, a mini-batch of samples of source domain with their noisy labels are sampled and denoted as $\{(X_i, Y_i^*)\}_{i=1}^N$. We denote the outputs of the classifier h for these samples as the $N \times C$ matrix O . Each row of O represents the output of a sample, and each column of O is a probability value over C categories. We denote the Y^* by a $0 - 1$ matrix L . Each row of L is a one-hot vector and represents the label of the corresponding sample. We have $\mathbf{T}_{h(X), Y^*} = OL/N$. Thus, $\text{DMI}(h(X), Y^*) = |\det(OL/N)|$.

To this end, the final loss function is defined as

$$L_{clf} = -\log \left(\left| \det \left(\frac{1}{N} OL \right) \right| \right) + \gamma L_r \quad (7)$$

where $\gamma > 0$ is the hyperparameter to weight the importance of the regularization item, the regularization item L_r is defined as

$$L_r = \frac{1}{N} \sum_{i=1}^N H(O_{i,:}) - H\left(\frac{1}{N} \sum_{j=1}^N O_{:,j}\right) \quad (8)$$

where the first term of equation 8 aims at forcing the output of h to be sparse enough to approximate an one hot vector, the second term of equation 8 aims at forcing the output of h to be equally distributed. The regularization item can also be interpreted as to constrain the source domain samples to be clustered into multiple clusters in a unsupervised manner.

4.2 The local preserving module

The local preserving module is to improve the robustness of RLPGA to label noise from the perspective of the learned feature representation. As we can see, the label noise of the source domain samples can cause a sample belonging to class i to be annotated into class j . From the Fig. 1, we can see that these noisy labels will influence the measurement of the classifier, e.g., the loss L_{clf} . During the gradient propagation stage, the feature representation will be learned by mistake. From a geometrically intuitive point of view, this mistake can be regarded as pushing a i -th class sample from an area surrounded by i -th class samples to an area surrounded by j -th class samples. Therefore, one solution to improve this problem is to design a method that can maintain the local structure of the original input space in the learned latent feature space. To this end, inspired by [48], we propose to excavate two kinds of relations. One is a similar relationship that should be preserved, the other is a dissimilar relationship that should be punished. We construct two weight matrices including an adjacency weight matrix and a negative weight graph to describe the above two relations.

Let $X^s = \{X_1^s, \dots, X_{m_s}^s\}$ be the source domain dataset, where X_i^s is a source domain sample and m_s is the total number of samples in the source domain dataset. Let $X^t = \{X_1^t, \dots, X_{m_t}^t\}$ be the target domain dataset, where X_i^t is a target domain sample and m_t is the total number of samples in the target domain dataset. Inspired by Locally Preserved Projection (LPP) [49], k -nearest-neighbor method is introduced to construct the adjacency graph. For the input space of the source domain, if X_j^s is one of the k nearest neighbors of X_i^s (or X_i^s is one of the k nearest neighbors of X_j^s), we build an edge between X_i^s and X_j^s . We traverse all the input samples, and then get the adjacency graph, which consists of the vertexes (the source domain samples) and the edges. An example of the adjacency graph is shown in Fig. 2 and we set $k = 5$. It is known that an instance and its k nearest neighbors are very likely to belong to the same category. Thus, the adjacency graph can represent the local similar relationship of the input dataset. We also define $H_{pos}^s \in R^{m_s \times m_s}$ as adjacency weight matrix. $H_{pos}^s(i, j)$ can be regarded as the similarity between X_i^s and X_j^s , which can be defined as: If $X_i^s \in N_k(X_j^s)$ or $X_j^s \in N_k(X_i^s)$,

$$H_{pos}^s(i, j) = \exp \left(\frac{-(d(X_i^s, X_j^s))^2}{t_1} \right) \quad (9)$$

otherwise, $H_{pos}^s(i, j) = 0$. $N_k(X_i^s)$ is the k nearest neigh-

bors of X_i^s , $d(\cdot)$ is a distance metric, and parameter t_1 is a prespecified hyperparameter. As we can see, the adjacency weight matrix H_{pos}^s can not only represent the connection state of any two input samples, but also the importance of any connected edge (or the similarity of any two input samples). Similarly, for target domain dataset, we can also obtain the corresponding adjacency weight matrix $H_{pos}^t \in R^{m_t \times m_t}$.

The negative weight matrix is to express the structure of dissimilar among input samples. We need to divide the samples of each domain into several clusters, and we suppose the samples that belong to different clusters are dissimilar. For source domain, if X_j^s is the nearest neighbors of X_i^s , we build an edge between X_i^s and X_j^s . We traverse all the input samples and finally construct a graph called negative graph, which consists of the vertexes (the source domain samples) and the edges. We definite that if there is a path between X_i^s and X_j^s , then X_i^s and X_j^s belonging to the same cluster. We traverse the obtained graph and finally construct several clusters. An example of obtained clusters is shown in Fig. 2 and we obtain 3 clusters. Let $\{1, 2, \dots, M\}$ be cluster number and M is the total number of clusters. Let $B \in \mathbb{R}^{m_s \times 1}$, and $B_i = m$, which means that sample X_i^s belongs to the cluster m , $m \in \{1, 2, \dots, M\}$. We also define the negative weight matrix as $H_{neg}^s \in R^{m_s \times m_s}$. $H_{neg}^s(i, j)$ represents dissimilarity between X_i^s and X_j^s , which can be defined as: If $B_i \neq B_j$, we have

$$H_{neg}^s(i, j) = \exp\left(\frac{-(d(X_i^s, X_j^s))^2}{t_1}\right) \quad (10)$$

otherwise, $H_{neg}^s(i, j) = 0$. The parameter t_1 is a prespecified hyperparameter. We can observe from the equation 10 that if X_i^s and X_j^s belong to different clusters, the closer the distance between them, the larger the value of $H_{neg}^s(i, j)$. This indicates that the boundary points of different clusters are important. Also, the negative weight matrix can not only indicate whether any two input samples belong to the same cluster, but also the dissimilarity of any two input samples. Similarly, as for target domain samples, we can also obtain a negative weight matrix $H_{neg}^t \in R^{m_t \times m_t}$.

To this end, let $H_{pn}^u = H_{pos}^u - H_{neg}^u$ and $u \in \{s, t\}$, the objective of the local preserving module is denoted as:

$$\begin{aligned} \min_f Dis_{pn}(f) = \\ \log\left(1 + \sum_{u \in \{s, t\}} \sum_{i, j=1}^{m_u} \exp\left(\|f(X_i^u) - f(X_j^u)\|_2^2 H_{pn}^u(i, j)\right)\right) \end{aligned} \quad (11)$$

Also, the choice of distance $d(\cdot)$ in equation 9 and 10 is based on different dataset. For higher dimensional data, we choose cosine distance, otherwise, choose Euclidean distance.

We can see that minimizing equation 11 can make points that are near in the original space project closer to the latent space, and points that belong to different clusters in the original space project farther into the latent space. This can be regarded as preserving the local structure including the similarity structure and dissimilar structure. Based on the prior knowledge that samples belonging to the same category are likely gathered together and samples belonging to different categories are likely separated from each other. So, minimizing equation 11 can enhance the feature discriminability. As the training process continues, noisy

label could make samples from some classes are mixed with samples from other classes. This can reduce the feature discriminability in the latent space. Thus, when aligning the distributions of two domains, some bad results will occur. However, the proposed local preserving method is to happen to force the features learned to be more discriminative, thereby improving the robustness of RLPGA to label noises.

4.3 The global aligning module

The global aligning module is to align the distributions between the source domain and target domain. The 'global' means the alignment operation is related to the whole distribution. Motivated by [8], we minimize the Wasserstein distance to align the two distributions. Specifically, for $\forall P_r, P_g \in \text{Prob}(X)$ and the corresponds support set Σ_r, Σ_g , the p th Wasserstein distance can be defined as

$$W_p(P_r, P_g) = \inf_{\varsigma(X_a, X_b) \in \Pi(X_a, X_b)} \int c(X_a, X_b)^p d\varsigma, \quad (12)$$

where $X_a \in \Sigma_r, X_b \in \Sigma_g$, $c(X_a, X_b)$ represents the distance of two patterns in Σ_r, Σ_g and $\Pi(X_a, X_b)$ denotes the set of all joint distributions $\varsigma(X_a, X_b)$ that satisfies $P_r = \int_y \varsigma(X_a, X_b) dX_b, P_g = \int_x \varsigma(X_a, X_b) dX_a$. Based on Kantorovich-Rubinstein theorem [50], the dual form of Wasserstein distance is written as

$$W_p(P_r, P_g) = \sup_{\|\vartheta\|_L \leq 1} E_{X_a \sim P_r} [\vartheta(X_a)] - E_{X_b \sim P_g} [\vartheta(X_b)], \quad (13)$$

where $\vartheta : X \rightarrow R$ is the 1-Lipschitz function and satisfies $\|\vartheta\|_L = \sup_{X_1 \neq X_2} |\vartheta(X_1) - \vartheta(X_2)| / |X_1 - X_2| \leq 1$. Also, the ϑ is called as the Critic and is implemented by an MLP.

To this end, the objective of the global aligning module is defined as,

$$\min_f W_p(P_s^{f(X)}, P_t^{f(X)}). \quad (14)$$

4.4 The final objective function

Based on Subsection 4.1, 4.2, and 4.3, the overall objective function of RLPGA is formulated as following, and the training process of RLPGA is shown in Algorithm ??.

$$\begin{aligned} \min_{f, h} L_{clf}(f, h) + \alpha Dis_{pn}(f) + \\ \beta W_p(P_s^{f(X)}, P_t^{f(X)}) + \Delta(f, h) \end{aligned} \quad (15)$$

4.5 Theoretical analysis

We provide some theoretical analysis about the robustness and target risk on our proposed RLPGA.

Lemma 4.1. (Properties of DMI [46]). DMI is non-negative, symmetric, and information-monotone. Moreover, it is relatively invariant: for random variables X_1, X_2 and X_3 , if X_3 is independent of X_2 conditioning X_1 , let $T_{X_1 \rightarrow X_3}$ be the matrix format of the joint distribution $\Pr(X_3 | X_1)$, then, the following holds

$$\text{DMI}(X_2, X_3) = \text{DMI}(X_2, X_1) |\det(T_{X_1 \rightarrow X_3})| \quad (16)$$

Theorem 4.1. For UDA with the noisy label, the proposed RLPGA is robust to label noise and the informative-theoretic-based loss is conducive to shrinking the upper bound of target risk R_{P_t} .

Algorithm 1 : RLPGA

Input: source data X^s , target data X^t , minibatch size m , the Critic learning rate α_1 in Wasserstein distance, the feature extractor learning rate α_2 , the classifier learning rate α_3 , hyperparameters α, β, γ , and the number of neighbor points k .

Initialize the neural network parameters of the feature extractor f , the classifier h , and the Critic ϑ with random weights $\omega_f, \omega_h, \omega_\vartheta$.

repeat

 Sample minibatch samples from X^s and X^t . Construct four weight matrices $H_{pos}^s, H_{neg}^s, H_{pos}^t, H_{neg}^t$.
for $t = 1$ **to** m **do**
 $Z^s \leftarrow f_{\omega_f}(X^s), Z^t \leftarrow f_{\omega_f}(X^t)$
 $\omega_{wd} \leftarrow \omega_{wd} + \alpha_1 \partial_{\omega_\vartheta} W_p$
end for
 $\omega_h \leftarrow \omega_h - \alpha_3 \partial_{\omega_h} L_{clf}$
 $\omega_f \leftarrow \omega_f - \alpha_2 \partial_{\omega_f} [L_{clf} + \alpha Dis_{pn} + \beta W_p + \Delta]$
until $\omega_f, \omega_h, \omega_\vartheta$ converge.

Proof: For objective 2, the first term, used to measure the performance of the classifier, is influenced by the noisy label, directly. From Lemma 4.1 , we can obtain

$$\text{DMI}(h(\cdot), Y^*) = \text{DMI}(h(\cdot), Y) |\det(T_{Y \rightarrow Y^*})| \quad (17)$$

where Y^* is the noisy version of the ground truth label Y . For every two classifiers h_1 and h_2 , we can see that the necessary and sufficient conditions for $\text{DMI}(h_1(\cdot), Y) > \text{DMI}(h_2(\cdot), Y)$ is $\text{DMI}(h_1(\cdot), Y^*) > \text{DMI}(h_2(\cdot), Y^*)$. In this paper, we propose to use the informative-theoretic-based loss to measure the performance of the classifier, we thus obtain that the necessary and sufficient conditions for $L_{clf}(f, h_1(\cdot), Y) > L_{clf}(f, h_2(\cdot), Y)$ is $L_{clf}(f, h_1(\cdot), Y^*) > L_{clf}(f, h_2(\cdot), Y^*)$. Therefore, we can obtain that the measurement based on noisy labels is consistent with the measurement based on clean labels. To this end, we conclude that RLPGA is robust to label noise.

Let h_1 be the learned classifier, from [8], we can know that the target error of the UDA can be bounded by the follows:

$$R_{P_t}(h_i) \leq R_{P_s}(h_i) + 2W_p(P_s^{f(X)}, P_t^{f(X)}) + \psi \quad (18)$$

As we can see, if the objective 2 is minimized, the first term $R_{P_s}(h_i)$ is equal to 0. When the label is the noisy version, for traditional loss function such as cross entropy loss, even if $R_{P_s}(h_i)$ is equal to 0, the value of $R_{P_s}(h_i)$ under the clean ground truth label is greater than 0. But for our proposed loss function, the result under the noisy labels is consistent with result under the clean labels. Thus, we can obtain that the upper bound of $R_{P_t}(f, h_1)$ is less than the upper bound of $R_{P_t}(f, h_2)$.

□

Proposition 4.1. When Z is a deterministic function of X , minimizing the equation 11 is also conducive to minimizing the target risk $R_{P_t}(f, h)$.

Proof: When Z is a deterministic function of X , we can obtain $\Pr(Z | X^u), u \in \{s, t\}$ is Dirac. Therefore, R_{P_t} can also be rewritten as

$$\begin{aligned} R_{P_t}(f, h) &= \int P_t^X(X) |h(f(X)) - Y^t| d(X) \\ &\doteq \int P_t^{f(X)}(Z) |h(Z) - Y^t| d(Z) \\ &= R_{P_s}(f, h) - \int P_s^{f(X)}(Z) |h(Z) - Y^s| d(Z) \\ &\quad + \int P_t^{f(X)}(Z) |h(Z) - Y^t| d(Z) \\ &= R_{P_s}(f, h) + \int P_t^{f(X)}(Z) (\Psi_t - \Psi_s) d(Z) \\ &\quad + \int (P_t^{f(X)}(Z) - P_s^{f(X)}(Z)) \Psi_s d(Z) \end{aligned} \quad (19)$$

where $\Psi_u(Z) = |h(Z) - Y^u|, u = \{s, t\}$. Compared equation 19 with equation 15, we can know that minimizing the first term and the third term in equation 15 corresponds to the minimizing first and third terms in equation 19. It is impossible to directly minimize the second term of equation 19 in equation 15. However, this paper mainly focuses on the covariance shift, so, we have $Y^s = Y^t$. The first term of equation 15 can increase the ability of the classifier h to correctly classify the source domain samples. Minimizing equation 11 can increase the feature discriminability of both source domain samples and target domain samples, and the label information is related to the feature discriminability. So, minimizing equation 11 is conducive to make the learned classifier h to predict the true label of the target domain samples. Therefore, we can conclude that minimizing equation 11 is conducive to minimizing the second term equation 19 thus is also conducive to minimizing target risk $R_{P_t}(f, h)$.

□

5 EXPERIMENTS

We conduct experiments on one synthetic data and four benchmark datasets. Also, due to space constraints, additional experiments of three datasets can be found in Appendix. Besides, we put the deepgoing analysis of the discrepancy metric and the training time analysis in Appendix. The main lines behind the experiment section are as follows. In Subsection 5.3, we take experiments on the synthetic dataset is to show that the gradient of our proposed model is stable and can converge during the training process under different scales of label noise. In Subsection 5.4, we first conducted conventional unsupervised domain adaptation experiments of transfer tasks on benchmark datasets, and the reported tables show the experimental results when the source domain samples have ground truth labels. This subsection is to verify that our method also has performance advantages in the absence of noise. In Subsection 5.5, we show the experimental results when the labels of source domain samples are polluted by different noise ratios. This is to verify that our method is not only suitable for learning with label noise, but also for learning without label noise. In Subsection 5.6, we show the experimental results of the ablation study. This is to verify that both the informative-theoretic-based loss and the local preserving module can improve the robustness to the label noise. In Subsection 5.7, we perform several experiments to study the influence of the hyper-parameters in our proposed RLPGA. In Subsection 5.9, we show the feature transferability and the feature

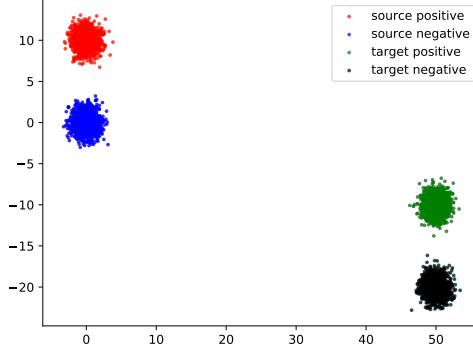


Fig. 3: Synthetic dataset

discriminability of our proposed RLPGA, intuitively. Also, in Appendix, we have given the experimental results in the case of random noise under the Digits dataset.

Synthetic dataset is the combination of source and target domain samples. Each domain includes 1000 positive samples and 1000 negative samples. The synthetic dataset is visualized in Fig. 3. We can see that the positive samples and negative samples of the source domain are represented by red dots and blue dots, and the positive samples and negative samples of the target domain are represented by green dots and black dots. **Office-Caltech10 dataset** [51] includes four domains, *e.g.*, Amazon (A), Webcam (W), DSLR (D), and Caltech (C). We adopt 4096-dimensional DeCaf feature [52] for the samples in Office+Caltech10 dataset. **Office31 dataset** [53] includes three domains, *e.g.*, Amazon (A), Webcam (W) and DSLR (D). **Office-Home dataset** [54] includes four domains, *e.g.*, Artistic images (Ar), Clip Art (Cl), Product images (Pr), and Real-World images (Rw). **Digits datasets** [19] includes two dataset including MINIST dataset and USPS dataset.

5.1 Compared Approaches

In this paper, we mainly compare our proposed RLPGA with both traditional learning methods and deep domain adaptation methods. The specific transfer task classification accuracy and average classification accuracy are reported. All experimental results of the compared methods in our submission are quoted from their respective original papers. To ensure the fairness and authenticity of the experimental results, we have not reproduced the experimental results of the compared methods on the data set that did not appear in the original article.

Traditional learning methods: Transfer Joint Matching (TJM) [55], Scatter Component Analysis (SCA) [56], Adaptation Regularization (ARTL) [57], Joint Geometrical and Statistical Alignment (JGSA) [58], CORrelation Alignment (CORAL) [59], Distribution Matching Machine (DMM) [55].

Deep domain adaptation methods: AlexNet [60], ResNet-50 [61], Deep Domain Confusion (DDC) [62], Deep Adaptation Network (DAN) [23], Maximum Mean Discrepancy Metric (MMD) [25], Domain Adversarial Neural Network (DANN) [19], Deep Correlation Alignment (DCORAL) [28], Adversarial Discriminative Domain Adaptation (ADDA) [63], Joint Adaptation Network (JAN) [31], Multi-Adversarial Domain Adaptation (MADA) [64], Similarity Network (SimNet) [65], Generate to Adapt (GTA)

[32], Deep Adversarial Attention Alignment (DAAA) [66], Conditional Domain Adversarial Network (CDAN) [67], Mainfold Embedded Distribution Alignment (MEDA) [68], Batch Spectral Penalization (BSP) [14], Wasserstein Distance Guided Representation Learning (WDGRL) [8], Contrastive Adaptation Network (CAN) [69], Certainty Attention based Domain Adaption (CADA) [70], Sliced Wasserstein Discrepancy (SWD) [33].

5.2 Implementation

All compared models in our experiments are implemented with TensorFlow and optimized by Adam optimizer. For each approach, all hyper-parameters are fixed, the batch size is set to 64 with 32 samples from each domain, the learning rate is set to 10^{-4} , the projection function f , the classifier h , and the Critic to approximate the Wasserstein distance are all set as MLP network, in which the activation function is set as Relu. Also, a softmax function is attached behind the classifier h to obtain a probabilistic output.

For the Synthetic dataset, the projection function f is approximated by a MLP network with one hidden layer of 20 nodes, the Critic in Wasserstein distance is approximated by a MLP network with two layers of 20 and 1 nodes. For Amazon and Email datasets, the projection function f is approximated by a MLP network with one hidden layer of 500 nodes, the Critic is approximated by a MLP network with two layers of 100 and 1 nodes. For Office-Caltech10, Office31, Office-Home, and Digits datasets, the projection function f is approximated by a MLP network with two hidden layers of 500 and 100 nodes, the Critic is approximated by a MLP network with two layers of 100 and 1 nodes. Note that the deep representations obtained by 50-layer ResNet are used as features for Office31, Office-Home, and Digits datasets.

To verify the robustness of our proposed RLPGA, we conduct experiments under different ratio noise. We list the explicit noise transition matrices as following, and for all experiments, r ranges from $\{0.2, 0.4, 0.6\}$. For Synthetic dataset, case (1):

$$P_{YY^*} = \begin{bmatrix} 1 & 0 \\ r & 1-r \end{bmatrix}. \quad (20)$$

For the rest of dataset, case (2):

$$\begin{bmatrix} 1 & & & & & \\ & \dots & & & & \\ r & & 1-r & & & r \\ & & & \dots & & \\ & & & & 1-r & r \\ & & & & & 1 \\ & & & & & & \dots \\ & & & & & & & 1-r \end{bmatrix} \quad (21)$$

There are four hyperparameters in our proposed RLPGA, that is α , β , γ , and k . For the synthetic dataset, we set $\alpha = 1$, $\beta = 0.1$, $\gamma = 1$, and $k = 3$. For Office-Caltech10 dataset, we set $\alpha = 1$, $\beta = 10$, $\gamma = 1$, and $k = 3$. For Office31 dataset, we set $\alpha = 1$, $\beta = 0.1$, $\gamma = 1$, and $k = 3$. For Office-Home dataset, we set $\alpha = 1$, $\beta = 10^3$, $\gamma = 1$, and $k = 3$. For Digits dataset, we set $\alpha = 1$, $\beta = 10$, $\gamma = 1$, and $k = 3$. For Email Spam Filtering dataset, we set $\alpha = 1$, $\beta = 10^{-2}$,

TABLE 1: Performance (accuracy) on Office+Caltech10 dataset with DeCaf features

Domains	A→C	A→D	A→W	W→A	W→D	W→C	D→A	D→W	D→C	C→A	C→W	C→D	Average
TJM	84.3	76.4	71.9	87.6	100	83.0	90.3	99.3	83.8	88.8	81.4	84.7	86.0
SCA	78.8	85.4	75.9	86.1	100	74.8	90.0	98.6	78.1	89.5	85.4	87.9	85.9
ARTL	87.4	85.4	88.5	92.3	100	88.2	92.7	100	87.3	92.4	87.8	86.6	90.7
JGSA	84.9	88.5	81.0	90.7	100	85.0	92.0	99.7	86.2	91.4	86.8	93.6	90.0
CORAL	83.2	84.1	74.6	81.2	100	75.5	85.5	99.3	76.8	92.0	80.0	84.7	84.7
DMM	84.8	92.4	84.7	86.5	98.7	81.7	90.7	99.3	83.3	92.4	87.5	90.4	89.4
AlexNet	83.0	87.4	79.5	83.8	100	73.0	87.1	97.7	79.0	91.9	83.7	87.1	86.1
DDC	85.0	89.0	86.1	84.9	100	78.0	89.5	98.2	81.1	91.9	85.4	88.8	88.2
DAN	84.1	91.7	91.8	92.1	100	81.2	90.0	98.5	80.3	92.0	90.6	89.3	90.1
MMD	88.6	90.5	91.6	92.2	100	88.6	90.1	98.9	87.8	93.1	91.6	91.2	92.0
DANN	87.8	82.5	77.8	82.9	100	81.3	84.7	98.9	82.1	93.3	89.5	91.2	87.7
DCORAL	86.2	91.2	90.5	88.4	100	88.6	85.8	97.9	85.4	93.0	92.6	89.5	90.8
MEDA	87.4	88.1	88.1	99.4	99.4	93.2	93.2	97.6	87.5	93.4	95.6	91.1	92.8
WDGRL	86.9	93.7	89.5	93.7	100	89.4	91.7	97.9	90.2	93.5	91.6	94.7	92.7
SWD	85.1	92.3	89.5	92.2	100	88.1	90.9	97.4	91.6	92.9	90.7	92.9	92.0
RLPGA	96.7	96.5	100	96.8	100	93.5	93.7	93.7	93.5	97.5	100	98.2	96.7

$\gamma = 0.1$, and $k = 3$. For Amazon Review dataset, we set $\alpha = 1$, $\beta = 1$, $\gamma = 10$, and $k = 3$.

5.3 Experimental results on synthetic dataset

We mainly compare our proposed RLPGA with RGA and WDGRL. Specifically, in RGA, the hyperparameter α is set to 0, which aims to eliminate the impact of the proposed two weight graphs. Fig. 4 shows the experimental results of the synthetic dataset for three methods including WDGRL, RGA, and our proposed RLPGA. Each row in Fig. 4 represents a different label noise ratio, and each column represents a different method. We record the value of the Wasserstein distance and the test classification accuracy during each iteration of the training process. As we can see, the Wasserstein distances in all three methods converge when step > 2000 , and the convergence curve is very smooth. So, we can conclude that all three methods show the gradient priority. Compared RGA with WDGRL, when the noise ratio is 0, we can see that the accuracy curve of RGA and WDGRL is almost the same. However, as the proportion of noise increases, the accuracy curve of WDGRL is obviously more oscillating than the accuracy curve of RGA. As the noise ratio goes to 0.6, the accuracy curve of RGA and WDGRL is almost the same again. This indicates that the proposed robust informative theoretic-based loss function is effective to reduce the impact of label noise to a certain degree. Compared RLPGA with WDGRL and RGA, we can see that the accuracy curve of RLPGA is obviously more stable than the accuracy curves of WDGRL and RGA under all noise ratios. In particular, in the case of all 4 noise ratios, the oscillation of the RLPGA accuracy curves is very weak, and the accuracy reaches 1. This demonstrates that the proposed two weight graphs are effective to reduce the impact of label noise and can promote the stability of the training process.

5.4 Conventional comparisons on transfer task

We conducted conventional unsupervised domain adaptation experiments of transfer tasks on benchmark datasets, and the reported tables show the experimental results when the source domain samples have ground truth labels. We

TABLE 2: Performance (accuracy) on Office31 dataset

Domains	A→D	A→W	D→A	D→W	W→A	W→D	Average
ResNet-50	68.9	68.4	62.5	96.7	60.7	99.3	76.1
DAN	78.6	80.5	63.6	97.1	62.8	99.6	80.4
DANN	79.7	82.0	68.2	96.9	67.4	99.1	82.2
ADDA	77.8	86.2	69.5	96.2	68.9	98.4	82.8
JAN	84.7	85.4	68.6	97.4	70.0	99.8	84.3
MADA	87.8	90.0	70.3	97.4	66.4	99.6	85.3
SimNet	85.2	88.6	73.4	98.2	71.6	99.7	86.1
GTA	87.7	89.5	72.8	97.9	71.4	99.8	86.5
DAAA	88.8	86.8	74.3	99.3	73.9	100.0	87.2
CDAN	93.4	93.1	71.0	98.6	70.3	100.0	87.7
MEDA	86.2	85.9	72.3	97.4	73.4	99.4	85.8
CAN	81.5	99.7	85.5	65.9	63.4	98.2	82.4
CADA	95.6	97.0	71.5	99.3	73.1	100.0	89.4
SWD	83.5	82.5	85.7	88.9	72.5	96.4	84.9
RLPGA	97.3	97.2	74.8	97.8	73.3	100.0	90.1

compare the performances of different methods based on specific transfer task classification accuracy and average classification accuracy.

1) *Comparisons on the Office+Caltech10 dataset with DeCaf features:* From Table 1, we observe that the average classification accuracy of RLPGA is 96.7%, which are 3.9% higher than the best among the 21 benchmark domain adaptation methods, especially, 4.0% higher than WDGRL, and 5.9% higher than DCORAL. As for specific transfer task classification accuracy, RLPGA achieves the best results on 10 specific transfer tasks. Also, we can observe that the best results among the 12 specific transfer tasks also appear in the deep domain adaptation methods compared with the traditional learning methods, and the best results are more likely to appear in the last line. The improvements of our Proposed RLPGA in this dataset are significant.

2) *Comparisons on the Office31 dataset:* From Table 2, we observe that the average classification accuracy of RLPGA is 90.1%, which is 0.7% higher than the best among the compared 14 domain adaptation methods, especially, 5.2% higher than SWD, and 7.9% higher than DCORAL. As for specific transfer tasks, RLPGA achieves the best results on 2 specific transfer tasks and obtains comparable results on 1 specific transfer task. Also, we observe that the results of domain adaptation methods are all better than the results of

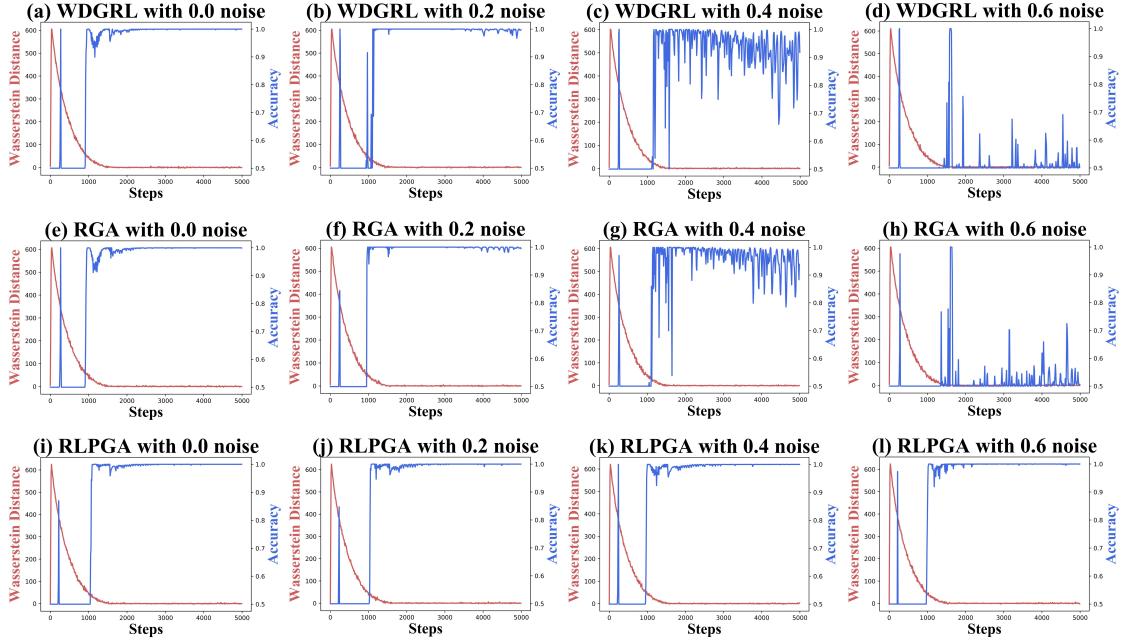


Fig. 4: The Wasserstein distance and test classification accuracy of WDGRL, RGA and RLPGA during each iteration
TABLE 3: Performance (accuracy) on Office-Home dataset

Domains	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Average
ResNet-50	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN	49.0	69.3	74.5	54.4	66.0	68.4	55.6	48.3	75.9	68.4	55.4	80.5	63.8
CDAN+E	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
BSP+DANN	51.4	68.3	75.9	56.0	67.8	68.8	57.0	49.6	75.8	70.4	57.1	80.6	64.9
BSP+CDAN	52.0	68.6	76.1	58.0	70.3	70.2	58.6	50.2	77.6	72.2	59.3	81.9	66.3
RLPGA	54.3	73.3	76.9	57.7	69.9	71.2	57.6	51.4	79.4	71.9	60.2	82.5	67.2

TABLE 4: Performance (accuracy) on Digits dataset

Domains	MNIST→USPS	USPS→MNIST	Average
DANN	90.4	94.7	92.6
ADDA	89.4	90.1	89.8
UNIT	96.0	93.6	94.8
CycADA	95.6	96.5	96.1
CDAN	93.9	96.9	95.4
CDAN+E	95.6	98.0	96.8
BSP+DANN	94.5	97.7	96.1
BSP+ADDA	93.3	94.5	93.9
BSP+CDAN	95.0	98.1	96.6
SWD	98.1	97.1	97.6
RLPGA	97.2	98.6	97.9

the source-only based method, i.e., ResNet-50 [61]. Overall, the improvements of our Proposed RLPGA in this dataset are not significant.

3) *Comparisons on the Office-Home dataset:* From Table 3, we can know that RLPGA achieves the best results on most tasks. For example, the average classification accuracy of RLPGA is 67.2%, which is 0.9% higher than the best among the compared 8 domain adaptation methods, 9.6% higher than DANN, and 21.1% higher than ResNet-50. As for specific transfer task classification accuracy, RLPGA achieves the best results on 8 of 12 specific transfer tasks and obtains

comparable results on the other 4 specific transfer tasks. The improvements of RLPGA in this dataset are significant.

4) *Comparisons on the Digits dataset:* From Table 4, we observe that the average accuracy of RLPGA outperforms all other methods. The experimental results are consistent with the comparisons of the previous experiments. The improvements of our Proposed RLPGA in this data are not significant.

In general, we can draw the following conclusions for the conventional comparisons on transfer task: 1) Deep domain adaptation methods are more effective than traditional learning methods; 2) The adversarial based methods are more effective than metric-based methods; 3) The learned latent feature representation of our proposed RLPGA is the most discriminative.

5.5 Denoising comparisons on transfer task

Fig. 5, 7, and 6 show the experimental results when the labels of source domain samples are polluted by different noise ratios. The hyper-parameter α of RGA is also set to 0, which aims to eliminate the impact of the two weight graphs. We compare the performances of different methods based on specific transfer task classification accuracy and average classification accuracy.

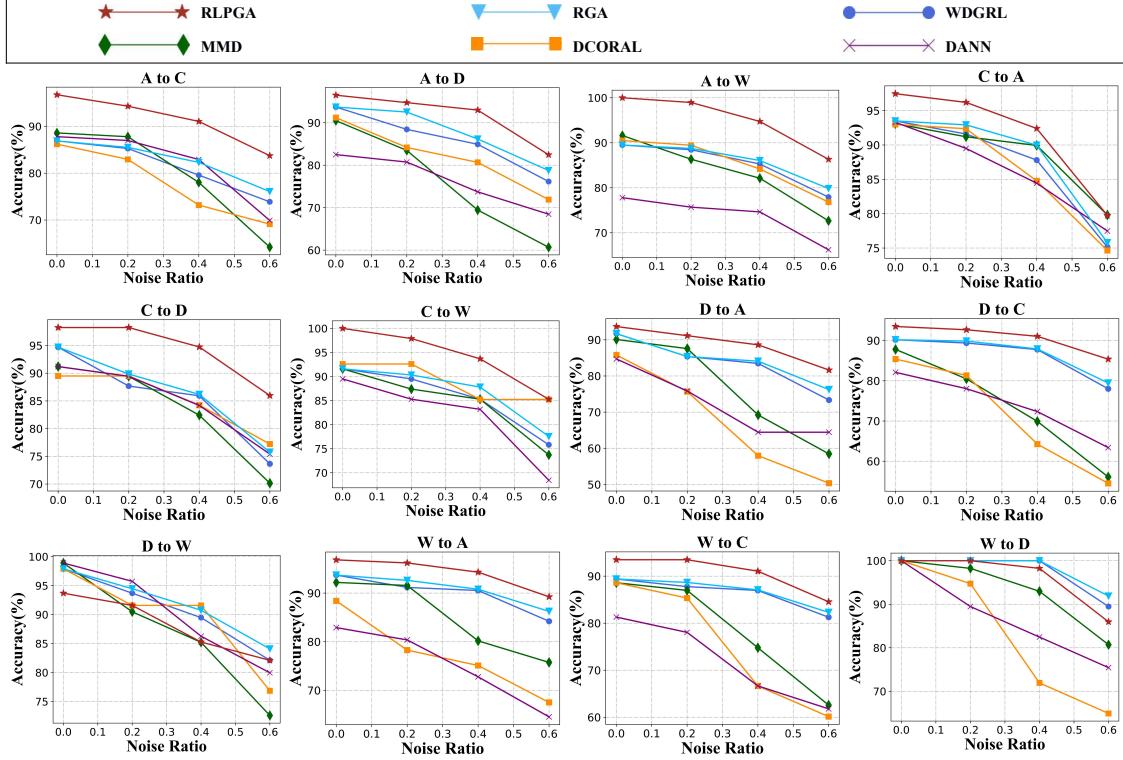


Fig. 5: Robustness evaluation on Office+Caltech10 dataset with DeCaf features

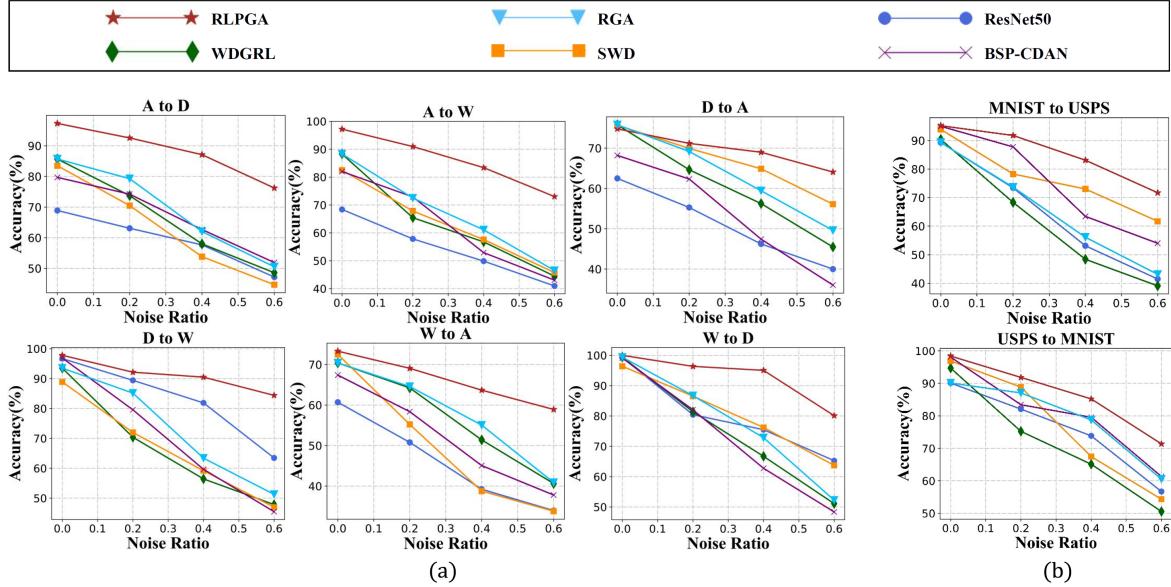


Fig. 6: Robustness evaluation on (a) Office31 dataset; (b) Digits dataset

1) Comparisons on the Office+Caltech10 dataset with DeCaf features: From Fig. 5, we observe that RLPGA achieves the best results under all noise ratios in 10 of 12 tasks. Especially, the resulting curve of RLPGA is the most stable and least decline with the addition of noise in all tasks.

2) Comparisons on the Office31 dataset: From Fig. 6 (a), we observe that RLPGA achieves the best results on 23 of 24 specific transfer tasks, and for A to D, A to W, D to W, W to A, and W to D transfer tasks, RLPGA achieves the best results on all different noise ratios. Also, when the noise

ratio is equal to 0.6, the accuracy of RLPGA is at least 10% higher than the other five methods on average.

3) Comparisons on the Digits dataset: From Fig. 6 (b), we observe that RLPGA achieves the best results on all specific transfer tasks. Especially, when the noise ratio is equal to 0.6, the accuracy of RLPGA is almost 10% higher than the other five methods on average. Also, the curve of RLPGA is smoother than other benchmark methods.

4) Comparisons on the Office-Home dataset: From Fig. 7, we observe that RLPGA achieves the best results on 34 of 48

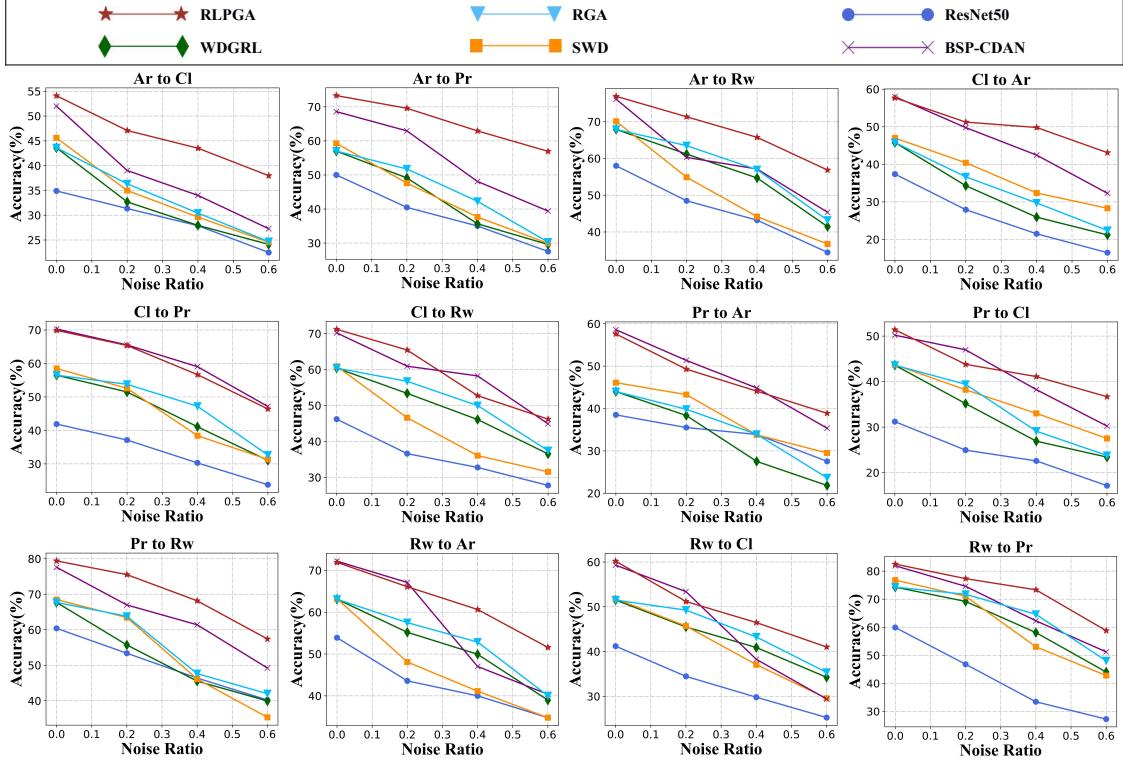


Fig. 7: Robustness evaluation on Office-Home dataset

specific transfer tasks. For specific transfer tasks, such as Ar to Cl, Ar to Pr, Ar to Rw, Pr to Rw, and Rw to Pr, RLPGA achieves the best results on 8 of 12 specific transfer tasks and obtains comparable results on other 4 specific transfer tasks. Also, when the noise ratio is equal to 0.6, the accuracy of RLPGA is higher than the other five methods on all specific transfer tasks.

Therefore, the denoising comparisons on the transfer task can thoroughly verify the robustness against label noise of our proposed RLPGA.

5.6 Ablation study

The proposed RLPGA is mainly composed of two parts including the proposed robust informative theoretic-based loss function and the constructed two adjacency weight matrices and two negative weight matrices to enhance the robustness to the noisy label. For ablation study, a simplified version of RLPGA, which does not use the two kind of weight matrices is verified and is named RGA. So, by evaluating the classification accuracy of RGA on different datasets under different noise ratio, we can verify whether the robust informative theoretic-based loss function is effective to improve the robustness. Comparing RLPGA with RGA, we can evaluate whether the constructed four weight matrices are effective to improve the robustness.

The main difference between RGA and WDGRl is that RGA adopts the proposed robust informative theoretic-based loss function to improve the robustness to label noise. Compared RGA with WDGRl, we observe that RGA outperforms WDGRl in most specific transfer tasks under different noise ratios of different data sets, where RGA achieves better results in 31 tasks out of all 48 tasks on

the Amazon review dataset, 42 tasks out of all 48 tasks on the Office-Caltech10 dataset with DeCaf features, 40 tasks out of 48 tasks on the Office-Caltech10 dataset with SURF features, 23 tasks out of 24 tasks on the Office31 dataset, 46 tasks out of 48 tasks on the Office-Home dataset, 10 tasks out of 12 tasks on the Email Spam Filtering dataset, and 7 tasks out of 8 tasks on the Digits datasets. Therefore, we can conclude that it is effective to consider the proposed robust informative theoretic-based loss function to reduce the sensitivity to label noise.

The main difference between RLPGA and RGA is that RLPGA constructs two weight graphs to enhance the feature discriminability, thereby reducing the influence of noise labels on the learned feature representation. We observe that RLPGA obtains better classification accuracy in many specific transfer tasks on different datasets, where RLPGA achieves better results in 41 tasks out of all 48 tasks on the Amazon review dataset, 44 tasks out of all 48 tasks on the Office-Caltech10 dataset with DeCaf features, 40 tasks out of 48 tasks on the Office-Caltech10 dataset with SURF features, 23 tasks out of 24 tasks on the Office31 dataset, all tasks on the Office-Home dataset, 11 tasks out of 12 tasks on the Email Spam Filtering dataset, and all tasks on the Digits datasets. Therefore, we can conclude that it is effective to reduce the influence of noise labels on the learned feature representation by enhancing the feature discriminability.

Note that many methods focus on adaptation architecture design. As for our proposed method, we mainly focus on the objective function, and the new network structures can be easily integrated into our framework. Even so, from the discussion in Subsection 5.2, we can know that the average accuracy of RLPGA outperforms most compared

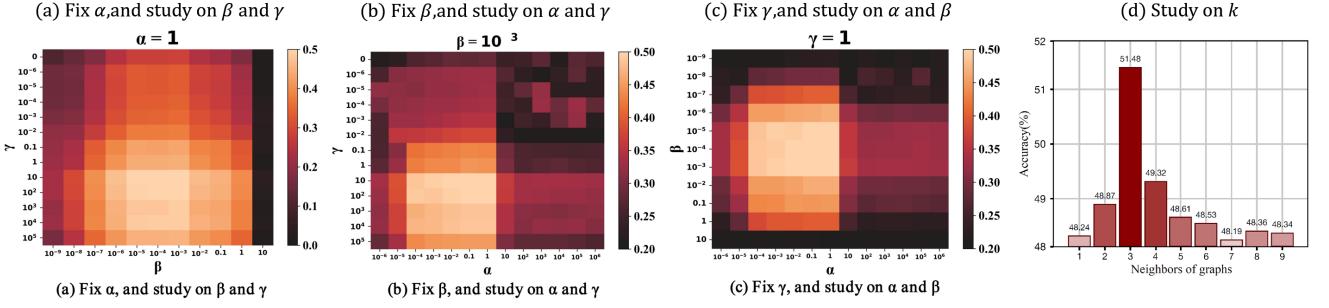


Fig. 8: The influence of hyper-parameters

methods on most datasets. Therefore, we can conclude that both the proposed robust informative theoretic-based loss and the constructed two weight graphs are effective to improve the robustness to label noise.

5.7 Influence of Hyper-parameters

Specifically, we performed several experiments to study the influence of the hyper-parameters in our proposed RLPGA including α which is used to balance the impact of the term $Dis_{pn}(f)$, β which is used to balance the impact of the term $W_p(P_s^{f(X)}, P_t^{f(X)})$, γ which is used to balance the impact of the term L_r . To intuitively understand the influences of the hyper-parameters, we take several experiments based on the transfer task Pr \rightarrow Cl of Office-Home dataset. As the results are shown in Fig. 8, the plots further elaborate our deepgoing studies' results with RLPGA. To explore the influence of α , we first fix $\beta = 10^3$, $\gamma = 1$ and $k = 3$ and then select the α from range of $\{10^{-6}, \dots, 10^{-1}, 1, 10^1, \dots, 10^6\}$. From the results, we observe that appropriate enhancement of feature discrimination can promote the performance of our proposed method. To explore the influence of β , we first fix $\alpha = 1$, $\gamma = 1$ and $k = 3$ and then select the β from range of $\{10^{-9}, \dots, 10^3\}$. From Fig. 8, we observe that the transferability of learned feature representation is important to the classification task. To explore the influence of γ , we first fix $\alpha = 1$, $\beta = 10^3$, $k = 3$, and then select the γ from range of $\{0, 10^{-5}, \dots, 1, \dots, 10^5\}$. From the results, we observe that the cross entropy loss and L_r are all important to the classification task.

We further conduct experiments based on the transfer task Pr \rightarrow Cl of Office-Home dataset to explore the influence of the number of neighbor points k , and the number of neighbor points k when constructing two weight matrices. We first fix $\alpha = 1$, $\beta = 10^3$, $\gamma = 1$, and then select the k from range of $\{1, \dots, 9\}$. The results are shown in Fig. 8 (d), we can see that an appropriate number of neighbor points is important.

5.8 The deepgoing analysis of the discrepancy metric

The time complexity of our proposed method will vary depending on the specific discrepancy metric taken, for instance, Wasserstein distance, KL divergence, etc. For the exact purpose of exploring the time complexity of RLPGA based on Wasserstein distance or KL divergence, we conduct several comparisons on the synthetic dataset. Fig. 10 shows the experimental results of time complexity, and in details,

TABLE 5: Performance (accuracy) on Digits dataset with different discrepancy metrics

Domains	MNIST \rightarrow USPS	USPS \rightarrow MNIST	Average
DANN	90.4	94.7	92.6
ADDA	89.4	90.1	89.8
UNIT	96.0	93.6	94.8
CyCADA	95.6	96.5	96.1
CDAN	93.9	96.9	95.4
CDAN+E	95.6	98.0	96.8
BSP+DANN	94.5	97.7	96.1
BSP+ADDA	93.3	94.5	93.9
BSP+CDAN	95.0	98.1	96.6
SWD	98.1	97.1	97.6
RLPGA w/ KL	93.6	95.3	94.5
RLPGA w/ WD	97.2	98.6	97.9

(a) represents the time complexity stats on the training stage of all time, (b) denotes the results on training stage after the convergence, and (c) represents the results on the testing stage of all time. The figure (a) actively demonstrates that in general, the time complexity of RLPGA with Wasserstein distance (i.e., RLPGA w/ WD) is higher than that of RLPGA with KL divergence (i.e., RLPGA w/ KL), but the difference is not significantly large. From figure (b), we observe that after convergence, the complexity of RLPGA w/ WD is more unstable than RLPGA w/ KL. As shown in figure (c), in the testing stage, the complexities of the compared methods are similar, to some degree.

Fig. 11 demonstrates the time complexity stats of the critic network on the training stage, and the subfigure (a) represents the records of all time, and the subfigure (b) denotes the records after convergence. From figure (a) of Fig. 10 and figure (a) of Fig. 11, we observe that compared with the main model of our proposed RLPGA, the critic network of the discrepancy metric reaches the convergence state faster. In addition, by observing Fig. 10 (b) and Fig. 11 (b), we find that after convergence, the fluctuation of time complexity is mainly brought by the critic network, and the particular reason for this circumstance is that with the entry of new batch of data, the critic network for the distribution discrepancy calculation of the source domain and the target domain will be updated all the way. However, after convergence, with the entry of a new batch of data, the main model of RLPGA will generate slight gradients, and the fluctuation of the optimization is accordingly trivial.

Along the lines of the experimental principle of Section 5.4, we further conduct experiments on the Digits dataset to

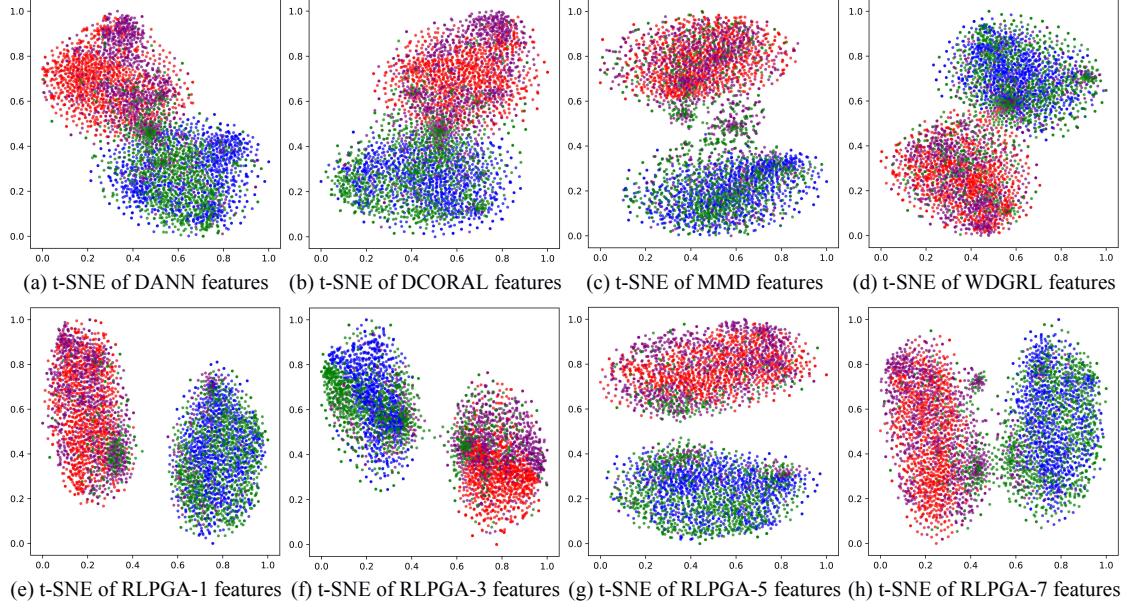
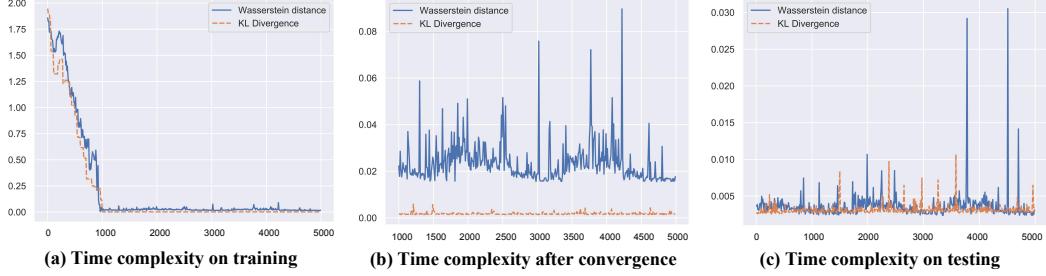
Fig. 9: Feature visualization of the $K \rightarrow E$ task in Amazon review dataset

Fig. 10: The analysis of time complexity on different discrepancy metrics

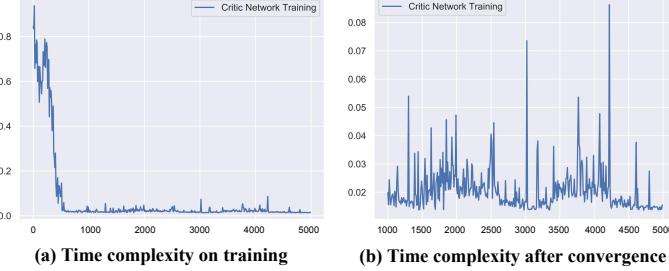


Fig. 11: The analysis of time complexity on the training stage of critic network

clarify the performance of RLPGA w/ WD and RLPGA w/ KL. As demonstrated in Table 5, we reckon that although the calculation of Wasserstein distance is more time-consuming than that of KL divergence, the former discrepancy metric can better depict the differences between the distributions of the source domain and the target domain. In detail, RLPGA w/ WD outperforms RLPGA w/ KL by 2.6% on MNIST→USPS task and 3.3% on USPS→MNIST task. Therefore, we adopt Wasserstein distance as the specific discrepancy metric for the proposed RLPGA.

5.9 Feature Visualization

To show the feature transferability and discriminability intuitively, we set the noise ratio r as 0.2 and visualize the features learned by the eight methods based on the $K \rightarrow E$

transfer task of Amazon review dataset. We introduce the t-SNE visualization to visualize the learned features and plot them in Fig. 9. For all subgraphs in Fig. 9, red and blue dots separately represent positive and negative samples in the source domain, and purple and green dots represent positive and negative samples in the target domain, respectively. High feature transferability should bring together dots of the same class in both domains, while high feature discriminability should separate dots of different classes from each other. We observe the feature transferability is learned well for all approaches. As for the feature discriminability, the representations learned by RLPGA outperform other approaches. So, this indicates that the proposed RLPGA is more effective and robust.

6 CONCLUSIONS

In this paper, we propose a novel method called robust local preserving and global aligning network for adversarial domain adaptation (RLPGA). RLPGA tackles the problem of learning domain adaptation models under the setting of noisy labels. First, RLPGA introduces a robust loss for solving this problem. We prove that it can reduce the impact of label noises. Then, to reduce the effect of label noises from the feature perspective, a local preserving and global aligning method is proposed. We also provide a theoretical analysis that RLPGA is conducive to minimize the target risk. Experiments results on sentiment and image classification domain adaptation datasets show the effectiveness of the proposed method. classification of RLPGA is 83.4%, which is 1% higher than the best among the 5 domain adaptation methods. For specific transfer task classification accuracy, RLPGA achieves the best results on 12 specific transfer tasks. The improvements of our Proposed RLPGA in this dataset are significant.

7 ACKNOWLEDGEMENTS

The authors would like to thank the associate editor and anonymous reviewers for their valuable comments. This work is supported in part by National Natural Science Foundation of China No. 61976206, No. 61832017, No. 91746301, No. 71531001, Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, CCF-Tencent Open Fund RAGR20200110, the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China 21XNLG05, Public Computing Cloud, Renmin University of China, Key Special Project for Introduced Talents Team of Southern Marine Science and Engineering Guangdong Laboratory (Guangzhou) No. GML2019ZD0603, and Strategic Priority Research Program of the Chinese Academy of Sciences Grant No. XDA19020500. This work is also supported in part by Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the “Double-First Class” Initiative, Renmin University of China, and Public Policy and Decision-making Research Lab of Renmin University of China.

REFERENCES

- [1] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, “Analysis of representations for domain adaptation,” in *Advances in neural information processing systems*, 2007, pp. 137–144.
- [2] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
- [3] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, “A comprehensive survey on transfer learning,” *Proceedings of the IEEE*, 2020.
- [4] S. Ben-David, T. Lu, T. Luu, and D. Pál, “Impossibility theorems for domain adaptation,” in *International Conference on Artificial Intelligence and Statistics*, 2010, pp. 129–136.
- [5] Niall and Adams, “Dataset shift in machine learning,” *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 2010.
- [6] H. Shimodaira, “Improving predictive inference under covariate shift by weighting the log-likelihood function,” *Journal of statistical planning and inference*, vol. 90, no. 2, pp. 227–244, 2000.
- [7] F. Wang, “Addressing two issues in machine learning: interpretability and dataset shift,” Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, USA, 2018.
- [8] J. Shen, Y. Qu, W. Zhang, and Y. Yu, “Wasserstein distance guided representation learning for domain adaptation,” *arXiv preprint arXiv:1707.01217*, 2017.
- [9] M. Chen, K. Q. Weinberger, and J. Blitzer, “Co-training for domain adaptation,” in *Advances in neural information processing systems*, 2011, pp. 2456–2464.
- [10] W.-S. Chu, F. De la Torre, and J. F. Cohn, “Selective transfer machine for personalized facial action unit detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3515–3522.
- [11] J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. J. Smola, “Correcting sample selection bias by unlabeled data,” in *Advances in neural information processing systems*, 2007, pp. 601–608.
- [12] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, “Deep domain confusion: Maximizing for domain invariance,” *arXiv preprint arXiv:1412.3474*, 2014.
- [13] K. Azizzadenesheli, A. Liu, F. Yang, and A. Anandkumar, “Regularized learning for domain adaptation under label shifts,” *arXiv preprint arXiv:1903.09734*, 2019.
- [14] X. Chen, S. Wang, M. Long, and J. Wang, “Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation,” in *International Conference on Machine Learning*, 2019, pp. 1081–1090.
- [15] L. Duan, D. Xu, and S.-F. Chang, “Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1338–1345.
- [16] A. Rozantsev, M. Salzmann, and P. Fua, “Beyond sharing weights for deep domain adaptation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 4, pp. 801–814, 2018.
- [17] K. You, X. Wang, M. Long, and M. Jordan, “Towards accurate model selection in deep unsupervised domain adaptation,” in *International Conference on Machine Learning*, 2019, pp. 7124–7133.
- [18] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, “Optimal transport for domain adaptation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 9, pp. 1853–1865, 2016.
- [19] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial training of neural networks,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.
- [20] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167–7176.
- [21] Y. Wu, E. Winston, D. Kaushik, and Z. Lipton, “Domain adaptation with asymmetrically-relaxed distribution alignment,” *arXiv preprint arXiv:1903.01689*, 2019.
- [22] H. Zhao, R. T. d. Combes, K. Zhang, and G. J. Gordon, “On learning invariant representation for domain adaptation,” *arXiv preprint arXiv:1901.09453*, 2019.
- [23] M. Long, Y. Cao, J. Wang, and M. I. Jordan, “Learning transferable features with deep adaptation networks,” *arXiv preprint arXiv:1502.02791*, 2015.
- [24] M. Long, H. Zhu, J. Wang, and M. I. Jordan, “Deep transfer learning with joint adaptation networks,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR.org, 2017, pp. 2208–2217.
- [25] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, “A kernel two-sample test,” *Journal of Machine Learning Research*, vol. 13, no. Mar, pp. 723–773, 2012.
- [26] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, “Deep domain confusion: Maximizing for domain invariance,” *arXiv preprint arXiv:1412.3474*, 2014.
- [27] B. Sun, J. Feng, and K. Saenko, “Return of frustratingly easy domain adaptation,” in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [28] B. Sun and K. Saenko, “Deep coral: Correlation alignment for deep domain adaptation,” in *European conference on computer vision*. Springer, 2016, pp. 443–450.
- [29] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
- [30] S. Kuroki, N. Charoenphakdee, H. Bao, J. Honda, I. Sato, and M. Sugiyama, “Unsupervised domain adaptation based on source-guided discrepancy,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 4122–4129.

- [31] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *International conference on machine learning*. PMLR, 2017, pp. 2208–2217.
- [32] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8503–8512.
- [33] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht, "Sliced wasserstein discrepancy for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10285–10295.
- [34] Z. Han, X.-J. Gui, C. Cui, and Y. Yin, "Towards accurate and robust domain adaptation under noisy environments," *arXiv preprint arXiv:2004.12529*, 2020.
- [35] Y. Shu, Z. Cao, M. Long, and J. Wang, "Transferable curriculum for weakly-supervised domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 4951–4958.
- [36] Y. Ge, D. Chen, and H. Li, "Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification," *arXiv preprint arXiv:2001.01526*, 2020.
- [37] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1-2, pp. 151–175, 2010.
- [38] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," *arXiv preprint arXiv:0902.3430*, 2009.
- [39] C. Cortes, M. Mohri, and A. Muñoz Medina, "Adaptation algorithm and theory based on generalized discrepancy," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 169–178.
- [40] P. Germain, A. Habrard, F. Laviolette, and E. Morvant, "A pac-bayesian approach for domain adaptation with specialization to linear classifiers," in *International conference on machine learning*, 2013, pp. 738–746.
- [41] M. Mohri and A. M. Medina, "New analysis and algorithm for learning with drifting distributions," in *International Conference on Algorithmic Learning Theory*. Springer, 2012, pp. 124–138.
- [42] Y. Zhang, T. Liu, M. Long, and M. I. Jordan, "Bridging theory and algorithm for domain adaptation," *arXiv preprint arXiv:1904.05801*, 2019.
- [43] S. Kuroki, N. Charoenphakdee, H. Bao, J. Honda, I. Sato, and M. Sugiyama, "Unsupervised domain adaptation based on source-guided discrepancy," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 4122–4129.
- [44] Q. Dou, C. Ouyang, C. Chen, H. Chen, and P.-A. Heng, "Unsupervised cross-modality domain adaptation of convnets for biomedical image segmentations with adversarial loss," *arXiv preprint arXiv:1804.10916*, 2018.
- [45] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Advances in neural information processing systems*, 2018, pp. 8778–8788.
- [46] Y. Kong, "Dominantly truthful multi-task peer prediction with a constant number of tasks," in *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2020, pp. 2398–2411.
- [47] Y. Xu, P. Cao, Y. Kong, and Y. Wang, "L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise," in *Advances in Neural Information Processing Systems*, 2019, pp. 6222–6233.
- [48] B. Li, P. Zhang, J. Zhang, and L. Jing, "Unsupervised double weight graphs based discriminant analysis for dimensionality reduction," *International Journal of Remote Sensing*, vol. 41, no. 6, pp. 2209–2238, 2020.
- [49] X. He and P. Niyogi, "Locality preserving projections," in *Advances in neural information processing systems*, 2004, pp. 153–160.
- [50] F. Santambrogio, "Optimal transport for applied mathematicians," *Birkhäuser, NY*, vol. 55, no. 58–63, p. 94, 2015.
- [51] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2066–2073.
- [52] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International conference on machine learning*, 2014, pp. 647–655.
- [53] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European conference on computer vision*. Springer, 2010, pp. 213–226.
- [54] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5018–5027.
- [55] Y. Cao, M. Long, and J. Wang, "Unsupervised domain adaptation with distribution matching machines," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [56] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, "Scatter component analysis: A unified framework for domain adaptation and domain generalization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 7, pp. 1414–1430, 2016.
- [57] M. Long, J. Wang, G. Ding, S. J. Pan, and S. Y. Philip, "Adaptation regularization: A general framework for transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1076–1089, 2013.
- [58] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1859–1867.
- [59] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [60] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [62] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014.
- [63] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7167–7176.
- [64] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," *arXiv preprint arXiv:1809.02176*, 2018.
- [65] P. O. Pinheiro, "Unsupervised domain adaptation with similarity learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8004–8013.
- [66] G. Kang, L. Zheng, Y. Yan, and Y. Yang, "Deep adversarial attention alignment for unsupervised domain adaptation: the benefit of target expectation maximization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 401–416.
- [67] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Advances in Neural Information Processing Systems*, 2018, pp. 1640–1650.
- [68] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, and P. S. Yu, "Visual domain adaptation with manifold embedded distribution alignment," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 402–410.
- [69] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4893–4902.
- [70] V. K. Kurmi, S. Kumar, and V. P. Namboodiri, "Attending to discriminative certainty for domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 491–500.
- [71] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Proceedings of the 45th annual meeting of the association of computational linguistics*, 2007, pp. 440–447.



Wenwen Qiang received the MS degree in the department of mathematics, college of science, China Agricultural University, Beijing, in 2018. He is currently a doctoral student at the University of Chinese Academy of Sciences. His research interests include transfer learning, deep learning, and machine learning.



Hui Xiong received his Ph.D. in Computer Science from the University of Minnesota - Twin Cities, USA, in 2005, the B.E. degree in Automation from the University of Science and Technology of China (USTC), Hefei, China, and the M.S. degree in Computer Science from the National University of Singapore (NUS), Singapore. He is a chair professor at the Hong Kong University of Science and Technology (Guangzhou). He is also a Distinguished Professor at Rutgers, the State University of New Jersey, where he

received the 2018 Ram Charan Management Practice Award as the Grand Prix winner from the Harvard Business Review, RBS Dean's Research Professorship (2016), two-year early promotion/tenure (2009), the Rutgers University Board of Trustees Research Fellowship for Scholarly Excellence (2009), the ICDM-2011 Best Research Paper Award (2011), the Junior Faculty Teaching Excellence Award (2007), Dean's Award for Meritorious Research (2010, 2011, 2013, 2015) at Rutgers Business School, the 2017 IEEE ICDM Outstanding Service Award (2017), and the AAAI-2021 Best Paper Award (2021). Dr. Xiong is also a Distinguished Guest Professor (Grand Master Chair Professor) at the University of Science and Technology of China (USTC). For his outstanding contributions to data mining and mobile computing, he was elected an ACM Distinguished Scientist in 2014, an IEEE Fellow and an AAAS Fellow in 2020. His general area of research is data and knowledge engineering, with a focus on developing effective and efficient data analysis techniques for emerging data intensive applications. He currently serves as a co-Editor-in-Chief of Encyclopedia of GIS (Springer) and an Associate Editor of IEEE Transactions on Data and Knowledge Engineering (TKDE), IEEE Transactions on Big Data (TBD), ACM Transactions on Knowledge Discovery from Data (TKDD) and ACM Transactions on Management Information Systems (TMIS). He has served regularly on the organization and program committees of numerous conferences, including as a Program Co-Chair of the Industrial and Government Track for the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), a Program Co-Chair for the IEEE 2013 International Conference on Data Mining (ICDM), a General Co-Chair for the IEEE 2015 International Conference on Data Mining (ICDM), and a Program Co-Chair of the Research Track for the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD2018).



Jiangmeng Li received the MS degree with concentration of data science, School of Professional Studies, New York University, New York, New York, USA, in 2018. He is currently a doctoral student at the University of Chinese Academy of Sciences. His research interests include transfer learning, deep learning, and machine learning.



Changwen Zhen received the Ph.D. degree in Huazhong University of Science and Technology. He is currently a professor in Institute of Software, Chinese Academy of Science. His research interests include computer graph and artificial intelligence.



Bing Su received the BS degree in information engineering from the Beijing Institute of Technology, Beijing, China, in 2010, and the PhD degree in electronic engineering from Tsinghua University, Beijing, China, in 2016. From 2016 to 2020, he worked with the Institute of Software, Chinese Academy of Sciences, Beijing. Currently, he is an associate professor with the Gaoling School of Artificial Intelligence, Renmin University of China. His research interests include pattern recognition, computer vision, and machine learning. He has published more than ten papers in journals and conferences such as IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on Image Processing (TIP), International Conference on Machine Learning (ICML), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE International Conference on Computer Vision (ICCV), etc.

8 APPENDIX

TABLE 6: Performance (accuracy) on Email dataset

Domains	$P \rightarrow u_1$	$P \rightarrow u_2$	$P \rightarrow u_3$	Average
MMD	81.0	86.0	94.1	87.0
DANN	83.3	85.7	91.9	86.9
DCORAL	79.7	83.8	89.8	84.4
WDGRL	85.7	88.3	95.8	89.9
SWD	87.2	88.8	94.5	90.2
RLPGA	87.6	89.1	97.1	91.3

8.1 Datasets

Amazon review dataset [71] records the product reviews on Amazon.com and includes four domains, *e.g.*, books (B), DVDs (D), electronics (E), and kitchen appliances (K). **Office-Caltech10 dataset** [51] includes four domains, *e.g.*, Amazon (A), Webcam (W), DSLR (D), and Caltech (C). We adopt 800-dimensional SURF feature for the samples in Office+Caltech10 dataset. **Email Spam Filtering dataset** [53] contains four user inboxes. We set the public inbox as the source domain and the other three private inboxes as target domains.

8.2 Extended conventional comparisons

We conducted extended conventional unsupervised domain adaptation experiments of transferring task on benchmark datasets, and the reported tables show the experimental results when the source domain samples have ground true labels. Note that only 5 baselines are presented in Table 6 and Table 7. The first reason is that all experimental results of the compared methods in our submission are quoted from their respective original papers. To ensure the fairness and authenticity of the experimental results, we have not reproduced the experimental results of the compared methods on the data set that did not appear in the original article. The second reason is that most of the compared methods are based on image data, the Amazon dataset and Email dataset are not an image dataset.

1) *Comparisons on the Email Spam Filtering dataset*: From Table 6, we observe that the best result always appears in the last column. RLPGA achieves the best average classification and reaches 91.3%, which is 1.4% higher than the best among the compared 4 domain adaptation methods and 7.9% higher than DCORAL. Also, RLPGA achieves the best classification accuracies on all specific transfer tasks. We can also observe that adversarial based methods such as RLPGA, WDGRL, and DANN are better than MMD and DCORAL. The improvements of our Proposed RLPGA in this dataset are significant.

2) *Comparisons on the Amazon review dataset*: From Table 7, we observe that the best result always appears in the last line. For specific transfer tasks, RLPGA achieves the best results on 12 specific transfer tasks. *E.g.*, the average classification of RLPGA is 83.4%, which is 1% higher than the best among the 5 domain adaptation methods. For specific transfer task classification accuracy, RLPGA achieves the best results on 12 specific transfer tasks. The improvements of our Proposed RLPGA in this dataset are significant.

3) *Comparisons on the Office+Caltech10 dataset with SURF features*: From Table 8, we observe that the proposed RLPGA has achieved the best results in more than half of the tasks. For specific transfer tasks, RLPGA achieves the best results on 6 of 12 specific transfer tasks and is the one that has achieved the best results the most times, *e.g.*, the average classification of RLPGA is 53.4%, which is 0.7% higher than the best result among the other 14 domain adaptation methods, 6.1% higher than WDGRL and, 7.3% higher than DCORAL. As for specific transfer task classification accuracy, RLPGA achieves the best results on 6 of 12 specific transfer tasks and is the one that has achieved the best results the most times. Also, we can conclude that the results of deep domain adaptation methods are overall better than the results of traditional learning methods. The improvements of our Proposed RLPGA in this dataset are significant.

8.3 Extended denoising comparisons

The figures show the extended experimental results when the labels of source domain samples are polluted by different noise ratios.

1) *Comparisons on the Email Spam Filtering dataset*: From Fig. 14, we observe that RLPGA achieves the best results on 10 of 12 specific transfer tasks. Especially, for P to u_1 and P to u_3 transfer tasks, RLPGA achieves the best results on all different noise ratios. Also, the curve of RLPGA has the least decline.

2) *Comparisons on the Amazon review dataset*: Fig. 12 shows the experimental results with different noise ratios. Compared RLPGA with the other 4 methods, we observe that RLPGA achieves the best results under all noise ratios in all tasks. Especially, when the noise ratio is equal to 0.6, the accuracy of RLPGA is at least 5% higher than the other five methods on average.

3) *Comparisons on the Office+Caltech10 dataset with SURF features*: From Fig. 13, we observe that RLPGA achieves the best results on 32 of 48 specific transfer tasks. Especially, for A to D, A to W, C to D, and C to W transfer tasks, RLPGA achieves the best results on all different noise ratios.

8.4 Denoising comparisons with random noise

In order to clarify the performance of our proposed method in different noise circumstances, we conduct extended experiments on Office-Home dataset when the labels of source domain samples are polluted by *random noise*. We set the noise rates in the range of {0, 0.1, 0.2, 0.3}, because the difficulty of remaining consistent performance under random noise is much higher than that of keeping robustness under the designed noise, which is based on explicit noise transition matrices of case (1) and (2).

Fig. 15 shows the experimental results on Office-Home dataset with different noise ratios (0, 0.1, 0.2, and 0.3). We compare RLPGA with the ablation model, *i.e.*, RGA, and the other 3 benchmark methods, *i.e.*, BSP+CDAN, SWD, and WDGRL. To understand the robustness of the compared methods, we further perform the backbone method, *i.e.*, ResNet50, and evaluate it within different noise rates. From the figure, we observe that under different noise ratios, RLPGA achieves the best results in most tasks. Especially,

TABLE 7: Performance (accuracy) on Amazon review dataset

Domains	B→D	B→E	B→K	D→B	D→E	D→K	E→B	E→D	E→K	K→B	K→D	K→E	Average
MMD	82.6	80.9	83.5	79.9	82.5	84.1	75.7	77.7	87.4	75.8	78.1	86.3	81.2
DANN	82.1	78.9	82.7	79.3	81.6	83.4	75.9	77.6	86.6	75.8	78.5	86.1	80.7
DCORAL	82.7	82.9	84.8	80.8	83.4	85.3	76.9	78.1	87.9	76.9	79.1	86.8	82.2
WDGRL	83.1	83.2	85.4	80.7	83.5	86.2	77.2	78.3	88.2	77.2	79.9	86.3	82.4
SWD	82.9	83.1	85.1	80.5	83.7	85.9	77.1	78.6	87.7	76.6	79.6	86.1	82.2
RLPGA	83.7	83.9	85.7	81.9	84.2	87.0	78.8	80.5	88.6	78.3	80.8	87.1	83.4

TABLE 8: Performance (accuracy) on Office+Caltech10 dataset with SURF features

Domains	A→C	A→D	A→W	W→A	W→D	W→C	D→A	D→W	D→C	C→A	C→W	C→D	Average
TJM	39.5	45.2	42.0	30.0	89.2	30.2	32.8	85.4	31.4	46.8	39.0	44.6	46.3
SCA	39.7	39.5	34.9	30	87.3	31.1	31.6	84.4	30.7	45.6	40.0	47.1	45.2
ARTL	36.1	36.9	33.6	38.3	87.9	29.7	34.9	88.5	30.5	44.1	31.5	39.5	44.3
JGSA	41.5	47.1	45.8	39.9	90.5	33.2	38.0	91.9	29.9	51.5	45.4	45.9	50.0
CORAL	45.1	39.5	44.4	36.0	86.6	33.7	37.7	84.7	33.8	52.1	46.4	45.9	48.8
MMD	44.1	41.4	37.3	34.1	84.7	30.7	32.5	73.6	30.7	54.8	40.3	47.1	45.9
DANN	45.0	41.4	38.6	34.1	82.8	32.7	31.6	74.2	32.2	54.9	43.4	47.8	46.6
DCORAL	45.0	40.1	38.3	34.9	84.1	33.3	31.5	73.9	31.5	53.4	40.0	47.1	46.1
WDGRL	45.9	44.6	40.7	32.2	81.5	31.1	35.6	77.0	32.6	55.2	42.4	48.4	47.3
SWD	44.1	42.3	40.5	32.2	80.4	30.6	32.9	75.6	33.1	54.8	41.9	48.9	46.4
MEDA	43.9	45.9	53.2	42.7	88.5	34.0	41.2	87.5	34.9	56.5	53.9	50.3	52.7
RLPGA	54.5	52.6	46.3	36.7	84.2	30.1	42.4	77.9	32.5	62.7	54.7	66.7	53.4

when the noise ratio is equal to 0.3, the accuracy of RLPGA is 3.8% higher than the best benchmark methods on average. We can find that the accuracies of the alternative methods are not very high under the random noise, but compared with other methods, our proposed RLPGA is still able to maintain the robustness to some extent.

8.5 Detailed influence of hyper-parameters

There are four hyper-parameters in our proposed RLPGA including α which is used to balance the impact of the term $Dis_{pos-neg}(f)$, β which is used to balance the impact of the term $W_p(P_s^{f(X)}, P_t^{f(X)})$, γ which is used to balance the impact of the term L_{RIT} , and the number of neighbor points k when constructing two weight matrices. To understand the influences of the four parameters intuitively, we take some experiments based on the transfer task Pr→Cl of Office-Home dataset. To explore the influence of α , we first fix $\beta = 10^{-3}$, $\gamma = 1$ and $k = 3$ and then select the α from range of $\{10^{-6}, \dots, 10^{-1}, 1, 10^1, \dots, 10^6\}$. The results are shown in Fig. 16. We observe that appropriate enhancement of feature discrimination can promote the performance of our proposed method. To explore the influence of β , we first fix $\alpha = 1$, $\gamma = 1$ and $k = 3$ and then select the β from range of $\{10^{-9}, \dots, 10^3\}$. From Fig. 16, we observe that the transferability of learned feature representation is important to the classification task. To explore the influence of γ , we first fix $\alpha = 1$, $\beta = 10^3$, $k = 3$, and then select the γ from range of $\{0, 10^{-5}, \dots, 1, \dots, 10^5\}$. From Fig. 16, we observe that the cross entropy loss and L_{RIT} are all important to the classification task. To explore the influence of the number of neighbor points k , we first fix $\alpha = 1$, $\beta = 10^3$, $\gamma = 1$, and then select the k from range of $\{1, \dots, 9\}$. The results are shown in Fig. 16, we can see that an appropriate number of neighbor points is important.

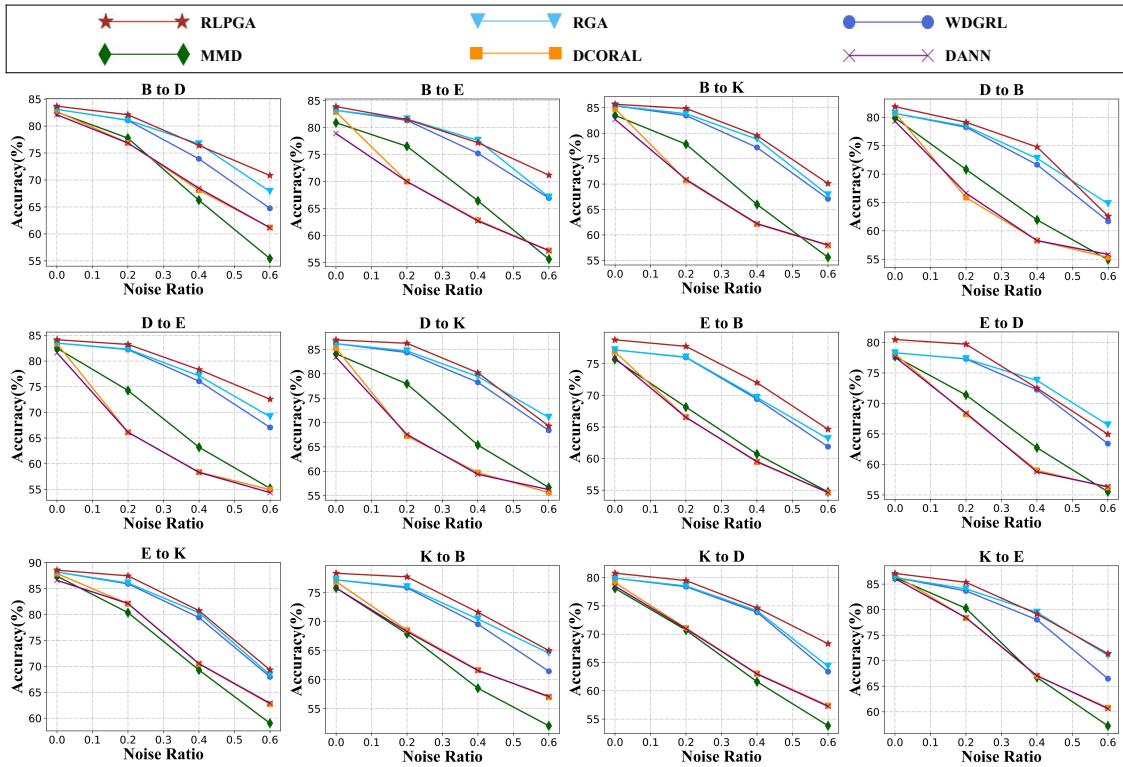


Fig. 12: Robustness evaluation on Amazon review dataset

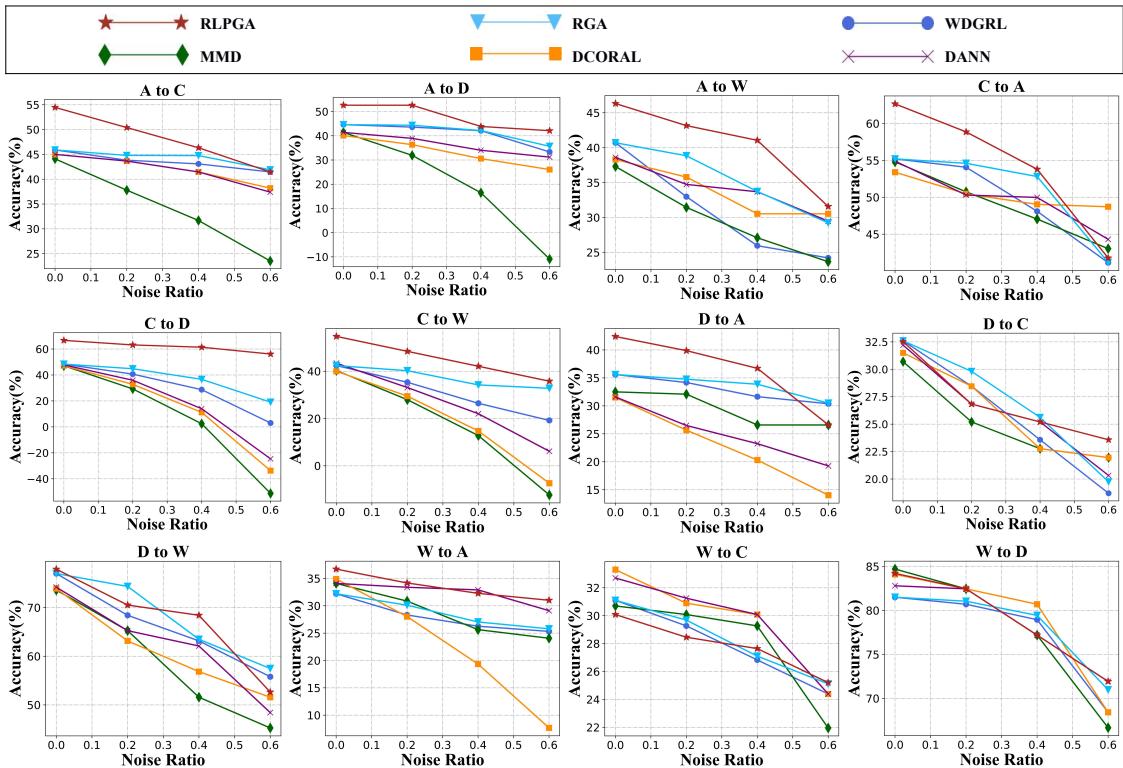


Fig. 13: Robustness evaluation on Office+Caltech10 dataset with SURF features

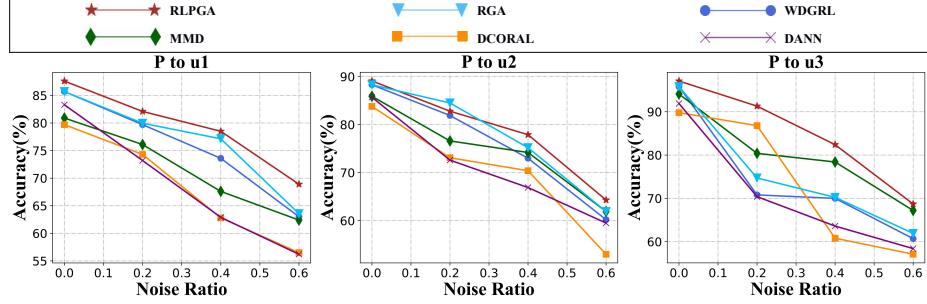


Fig. 14: Robustness evaluation on Email Spam Filtering dataset

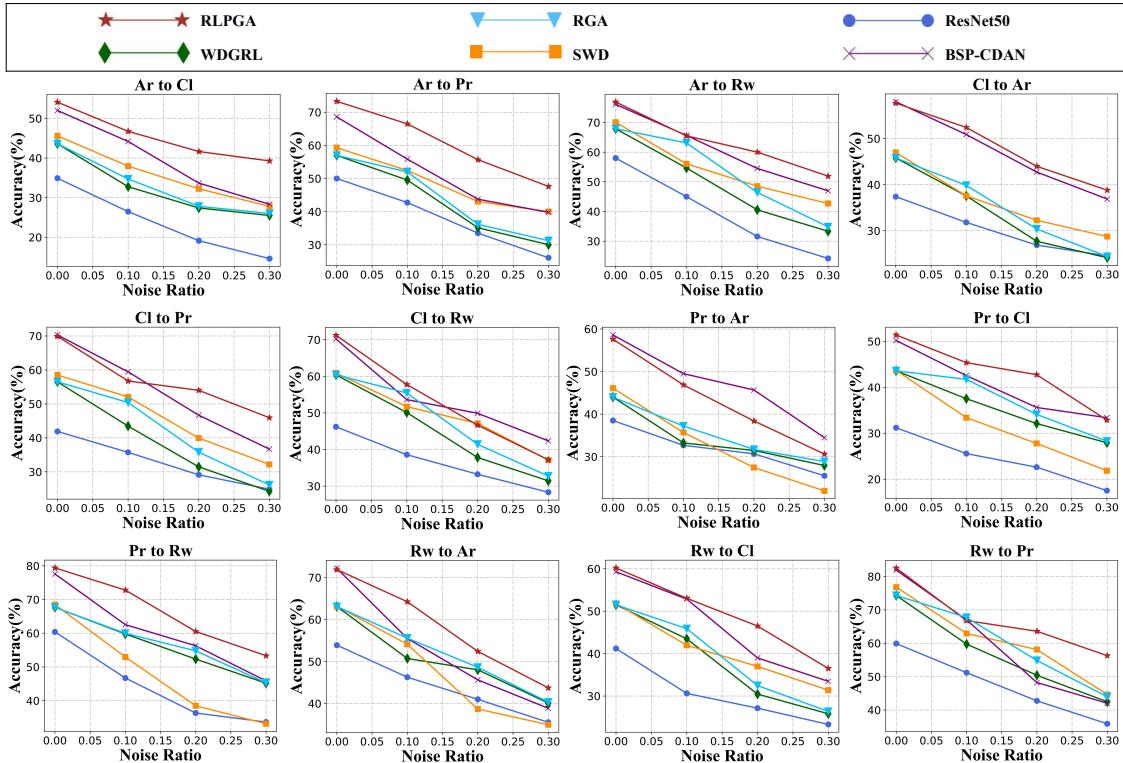


Fig. 15: Robustness evaluation on Office-Home dataset with random noise

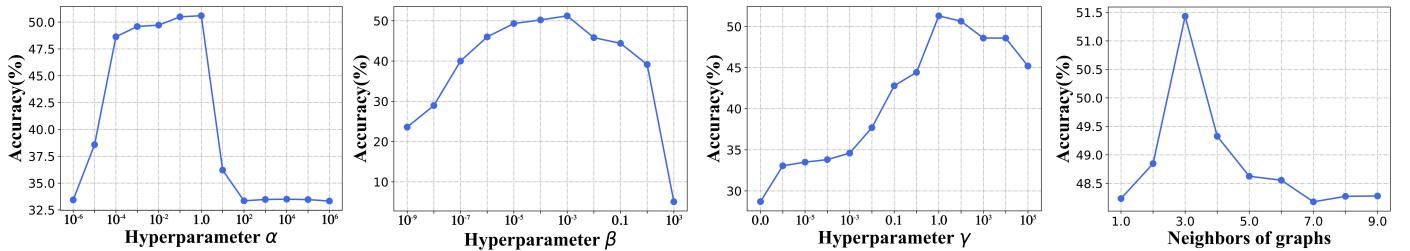


Fig. 16: The influence of parameters